THE SAMPLING CEILING: WHY TEST-TIME SELF-CONSISTENCY SATURATES AND AN ORACLE RULE FOR ALLOCATING SAMPLES

Anonymous authors*
Paper under double-blind review

ABSTRACT

Self-consistency can sample a correct answer and still vote for a wrong one. We study how confidence in the current answer differs from the value of drawing more samples. In modular addition, 40 samples contain a correct answer on 95.34% of instances, yet voting accuracy converges to 63.32% (three-seed means). In our *systematic-stochastic mixture*, voting can remove random errors on some instances, while on the rest the model always repeats one wrong answer. Each extra pair of samples then adds less majority-vote accuracy than the last, so a greedy oracle allocation is optimal for equally weighted queries with known per-query mixture parameters. When these are unknown, *Correlation-Aware Self-Consistency* (CASC) uses labeled population data to score the current answer's correctness and the risk of a systematic error. On public Grade School Math 8K (GSM8K) answer sets, the resulting tradeoffs depend on the model: default CASC uses 3.67 answers versus Early-stopping Self-Consistency's 8.24 on GPT-4, with a paired 95% interval for the accuracy difference that includes zero. Removing CASC's systematic-error term leaves its GPT-4 predictions and stopping times unchanged; Adaptive-Consistency's Beta rule is more accurate and cheaper on GPT-3.5. The oracle ranks queries by the gain from additional samples, and the stopping comparisons show that scores for the current answer do not measure that gain.

1 INTRODUCTION

Self-consistency (SC) answers a question by sampling multiple model outputs and returning their most frequent answer, the plurality vote (Wang et al., 2023). More draws can correct an unlucky first attempt, but they can also strengthen an incorrect consensus. When the model's most probable answer (its mode) is wrong, more samples make the vote return it ever more reliably. With a limited inference budget, one therefore needs to know when additional samples improve correctness and when they only make the model's preference more certain.

In modular addition, the task is to predict the sum of two integers modulo 59. At training step 1,800, a correct answer appears among 40 samples on 95.34% of instances, yet their plurality vote is correct on only 62.30% (three-seed means), and the model's complete answer probabilities show that infinitely many votes would reach only 63.32% (§5.3). *Answer availability* (the fraction of instances with a correct sample), finite-vote accuracy and the accuracy of the model's most probable answer thus differ. When each instance has a unique mode, the third is the *voting limit*, the accuracy that plurality voting approaches as samples grow.

Existing stopping rules rely on agreement. The Beta rule in Adaptive-Consistency (ASC/Beta; Aggarwal et al., 2023) compares the current leader with its runner-up, while Early-stopping Self-Consistency (ESC; Li et al., 2024) stops once all answers in a window of consecutive samples agree. Even a valid certificate of the model's most likely answer (Cordero-Encinar & Duncan, 2025; Ota et al., 2026) says nothing about whether that answer is correct. To judge the value of another sample, one must relate the vote to task correctness and not only to answer agreement.

We make this link with a *systematic-stochastic mixture* (SSM). On *recoverable* instances, each sample is independently correct with probability above one-half; on *systematic* instances, every sample repeats one wrong answer, the *lure*. Counting all wrong answers as one rival, SSM gives the exact

*Code: `https://anonymous.4open.science/r/CASC-E42F/`

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

majority-vote accuracy at every sample size, which rises to the recoverable fraction (§3). Differences between instances also make correctness correlated across the population, although each prompt's draws are independent. The voting limit above does not require SSM.

Queries with equal accuracy can benefit very differently from more samples. Two query types (SSM parameter settings) can both have 60% single-sample accuracy, yet gain 4.8 and 7.2 percentage points from two more votes (§4). Because gains shrink with each added pair of samples, an oracle that gives each pair to the query with the largest remaining gain is optimal for equally weighted queries with known error parameters and with sample counts fixed before any answer is seen.

For unknown query parameters, we develop *Correlation-Aware Self-Consistency* (CASC), which decides after each sample whether to stop (§4.3). Using an SSM fitted to labeled data, it scores the current answer's correctness and the risk of a systematic error, discounting *unanimous prefixes* (streams whose answers all agree so far) because they may come from systematic instances. These scores decide when to answer, sample again or abstain, but do not estimate the oracle's marginal gain.

We evaluate on two kinds of data. Modular-arithmetic transformers expose complete answer distributions (Power et al., 2022; Nanda et al., 2023), so their voting limits can be measured directly. Public Grade School Math 8K (GSM8K) answer banks (Cobbe et al., 2021), released sets of 40 sampled answers per question from three language models, extend the stopping comparison to text. CASC saves samples at some settings, while its systematic-error term mainly delays decisions. On arithmetic, mapping each method's scores to calibration-set accuracy largely removes the calibration gap between CASC and ASC/Beta without changing any decision (§5.4). Current correctness and the gain from further sampling should therefore be treated separately.

2 Related Work

Sampling, selection and ceilings. Self-consistency (Wang et al., 2023) aggregates sampled chains of thought (Wei et al., 2022). Repeated sampling can raise coverage (the chance that some sample is correct) while the accuracy of the selected answer plateaus (Brown et al., 2024); Bay & Yearick (2026) study ceilings set by the mode and by correlation and their implications for allocation. For one latent mixture, we derive exact correctness gains and an oracle allocation under known query parameters. Our stopping comparisons separate correctness calibration from the value of more samples.

Adaptive sampling and allocation. ASC/Beta (Aggarwal et al., 2023) and ESC (Li et al., 2024) stop from observed agreement; Difficulty-Adaptive Self-Consistency (DSC; Wang et al., 2025) also uses question difficulty. Confidence-Informed Self-Consistency (CISC; Taubenfeld et al., 2025) weights votes by response confidence, and Reliability-Aware Adaptive Self-Consistency (ReASC; Kim et al., 2026a) combines confidence with frequency. PETS (Liu et al., 2026) allocates samples so as to reproduce the vote of an infinite budget; our oracle instead maximizes correctness against the true answer under SSM. Verifier-guided allocation (Snell et al., 2024) and learned self-calibration (Huang et al., 2025b) use additional signals or training. Our comparisons hold the model and data fixed across SC, ASC/Beta, entropy stopping, ESC, Greedy and CASC.

Mode certificates and correctness. Certified Self-Consistency (Cordero-Encinar & Duncan, 2025) and Certification by Intersection-union Testing with E-processes (CITE; Ota et al., 2026) give statistical guarantees, under adaptive stopping, that the returned answer is the model's mode. A certified mode may still be wrong. Correctness calibration (Wang et al., 2024) and selective answering (Jurayj et al., 2025) therefore target correctness itself. Our calibration and abstention controls give all methods the same labels and evaluate calibration and selective accuracy separately. Appendix G discusses recent work on verification, reasoning depth and inference-time objectives.

Grokking as a mechanistic testbed. Grokking, in which test accuracy rises long after the training set is fit (Power et al., 2022), together with progress measures (Nanda et al., 2023) and representation-learning analyses (Liu et al., 2022), motivates our arithmetic testbed. Its checkpoints span a range of capability, expose the complete answer distribution, and provide exact labels.

3 A MIXTURE OF SYSTEMATIC AND STOCHASTIC ERRORS

3.1 SETUP

Draw an instance x from a population, with true answer $y^\star$ in an answer set of size M, and fix a model π. Repeated temperature sampling from the model produces the answer stream $Y_{1:n}$. A hidden state $S \in \{0, 1\}$ is drawn once per instance and shared by all of its samples; $\alpha \in [0, 1)$ is the probability of the systematic state $S = 0$.

Definition 1 (SSM). *With probability $1 - \alpha$ the instance is* recoverable *($S = 1$): samples are conditionally independent and identically distributed (i.i.d.), each correct with probability $q \in (\frac{1}{2}, 1)$; on error the answer follows a distribution over the $M - 1$ incorrect options. With probability α the instance is* systematic *($S = 0$): the model always returns one instance-specific wrong answer $\ell(x)$, the* lure*, so $Y_t = \ell(x)$ for every t with probability one.*

Here α is the fraction of instances that can never produce a correct sample. The fraction of finite streams with no correct answer estimates a different quantity, since a finite stream can miss the correct answer on a recoverable instance (Appendix B).

3.2 EXACT SAMPLING CURVE, SATURATION CEILING, AND CORRELATION

A majority requires more than half the votes; a plurality over answer categories (*categorical plurality*) requires only the largest count. The two differ when wrong answers split across categories. To analyze voting gains, we merge all wrong answers into one category and break ties by a fair coin; the correct answer then wins with probability $B(n, q) = \Pr(\mathrm{Bin}(n, q) > n/2) + \frac{1}{2}\mathbf{1}\{n \text{ even}\} \Pr(\mathrm{Bin}(n, q) = n/2)$, the *binary-majority benchmark*.

Theorem 1 (Sampling ceiling). *Under the SSM, binary-majority accuracy is*

$$A(n) = (1 - \alpha)B(n, q), \qquad \lim_{n\to\infty} A(n) = 1 - \alpha. \tag{1}$$

Its gap to the ceiling is at most $(1 - \alpha)e^{-2n(q-1/2)^2}$. For categorical plurality and odd n, $A(n)$ is a lower bound on accuracy, with equality when all recoverable errors are the same wrong answer. Categorical plurality also converges to $1 - \alpha$, since $q > 1/2$ makes the correct answer the unique mode in every recoverable instance.

The proof appears in Appendix A.1. Finite-sample plurality accuracy generally depends on the wrong-answer distribution as well as q. If each instance instead has its own random correctness probability q_i, binary-majority accuracy converges to $\Pr(q_i > 1/2) + \frac{1}{2}\Pr(q_i = 1/2)$. When every instance has a unique mode, the plurality limit is the fraction of instances whose mode is correct (Appendix E.1); finite-n accuracy can exceed this limit.

Pairwise correlation and effective sample size. Let $X_t = \mathbf{1}\{Y_t = y^\star\}$ with marginal $p = \mathbb{E}X_t = (1 - \alpha)q$. Distinct samples satisfy $\mathbb{E}[X_t X_u] = (1 - \alpha)q^2$, hence

$$\rho = \mathrm{Corr}(X_t, X_u) = \frac{\alpha q}{1 - (1 - \alpha)q} = \frac{\alpha q}{1 - p} \geq 0, \tag{2}$$

where the expectation is over both instances and their samples. For $\bar{X}_n = n^{-1}\sum_t X_t$, exchangeability gives

$$\mathrm{Var}(\bar{X}_n) = p(1 - p)\left[\tfrac{1-\rho}{n} + \rho\right] = \frac{p(1 - p)}{n_{\text{eff}}}, \qquad n_{\text{eff}} = \frac{n}{1 + (n - 1)\rho}, \tag{3}$$

This effective sample size is the number of independent samples that would give the same variance. For $\rho > 0$ it tends to $1/\rho$; for $\rho = 0$ it equals n. At $(\alpha, q) = (.25, .70)$, $n_{\text{eff}}(21) = 2.51$ (§5). This correlation comes only from differences between instances.

Proposition 1 (Identifiability). *For a nondegenerate SSM with $0 < p < 1$, labeled population moments (p, ρ) uniquely determine*

$$q = p + \rho(1 - p), \qquad \alpha = \frac{\rho(1 - p)}{p + \rho(1 - p)}.$$

These identities hold for the population; CASC uses finite-stream estimates (Appendix B). A fit to two moments does not show that the data follow SSM; Appendix B.1 gives a third-moment test.

3.3 WHY CONSENSUS CONFIDENCE NEED NOT PREDICT CORRECTNESS

Let v_1, v_2 be the counts of the two most frequent answers. Adaptive-Consistency's Beta rule (Aggarwal et al., 2023) models the leader's share within this pair with a $\mathrm{Beta}(v_1+1, v_2+1)$ distribution. Its score $c_n = I_{1/2}(v_2+1, v_1+1)$ is the probability that this share exceeds $1/2$.[1] Sampling stops when $c_n \geq c_{\mathrm{thr}}$ (default 0.95).

Proposition 2 (Beta confidence on systematic instances)**.** *On a systematic instance ($S=0$), every sample equals the lure, so $v_1 = n$, $v_2 = 0$ and reported confidence after n samples is*

$$c_n = I_{1/2}(1, n+1) = 1 - 2^{-(n+1)}. \tag{4}$$

It first crosses c_{thr} at $n^\star = \lceil \log_2 \frac{1}{1-c_{\mathrm{thr}}} \rceil - 1$ samples ($n^\star = 4$ at 0.95), committing to an answer that is wrong with probability one at nominal confidence $\geq c_{\mathrm{thr}}$ (for $c_{\mathrm{thr}} \in (1/2, 1)$ and a cap at least $n^\star$).

Unanimity has two explanations. A stream of identical answers fits a correct consensus ($S=1$, high q) and also a systematic instance ($S=0$). A posterior over answer frequencies can confirm that the answers are concentrated but says nothing about whether they are correct. Consider a symmetric SSM with uniform priors on the correct answer and the lure, and uniform recoverable errors. After a unanimous stream, the posterior odds that the instance is systematic are $\alpha/[(1-\alpha)(q^n + (1-q)^n/(M-1)^{n-1})]$. For $0 < \alpha < 1$ and $q < 1$, these odds grow without bound as n grows. CASC approximates this signal, which needs a population model as well as the observed unanimity.

4 FROM SAMPLING GAINS TO STOPPING DECISIONS

Here an oracle knows each query's mixture parameters (α_i, q_i) but not whether the query is actually recoverable or systematic. It chooses odd sample counts before observing answers and maximizes expected binary-majority accuracy. §4.3 treats adaptive stopping with unknown parameters.

4.1 THE GAIN FROM TWO MORE SAMPLES

Theorem 2 (Diminishing returns and peak gain)**.** *For $k \geq 1$, the gain from increasing $n = 2k-1$ to $n+2 = 2k+1$ is*

$$G_k(\alpha, q) = (1-\alpha)\binom{2k}{k}(q - \tfrac{1}{2})[q(1-q)]^k. \tag{5}$$

These positive gains decrease with k, since

$$\frac{G_{k+1}}{G_k} = \frac{2(2k+1)}{k+1} q(1-q) < 4q(1-q) < 1. \tag{6}$$

For fixed k and $\alpha < 1$, the gain is maximized at $q_k^\star = \frac{1}{2} + 1/(2\sqrt{2k+1})$. It vanishes as $q \downarrow 1/2$, $q \uparrow 1$, or $\alpha \uparrow 1$. As k grows, the accuracy $q_k^\star$ with the largest gain therefore moves toward the decision boundary $1/2$ without reaching it.

With fair ties, $A(2k) = A(2k-1)$, so samples are added in pairs. Equation 5 is the product of the recoverable mass $1-\alpha$, the tie probability $\binom{2k}{k}[q(1-q)]^k$, and the advantage $q - 1/2$ of the deciding vote. Near chance, this advantage vanishes; near certainty, ties are rare. The gain therefore peaks at an intermediate accuracy (derivation in Appendix A.2). Current accuracy does not determine the value of more samples; types $(\alpha, q) = (0, .6)$ and $(1/7, .7)$ both have single-draw accuracy .6, but gain .048 and .072 from two more votes.

4.2 OPTIMAL POPULATION ALLOCATION

Consider N equally weighted query types, each with its own known mixture parameters, and a total integer budget $T \geq N$. Choose odd counts $n_i \geq 1$ (optionally bounded by a common odd cap) to maximize $N^{-1}\sum_i A(n_i; \alpha_i, q_i)$ subject to $\sum_i n_i \leq T$.

[1] $I_z(a, b)$ is the regularized incomplete beta function.

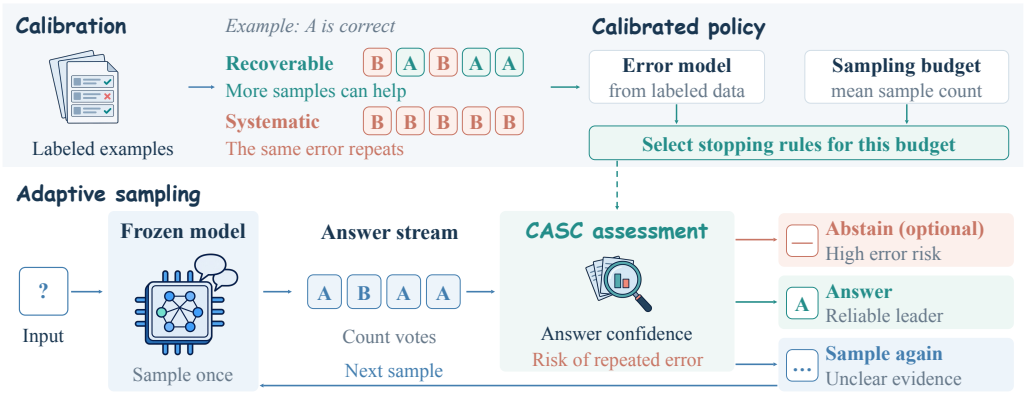


Figure 1: CASC overview. Calibration fits the error model and selects stopping rules for a target budget. The language model supplies sampled answers; after each one, CASC updates its confidence in the leading answer and its systematic-error risk, and then answers, samples again, or (optionally) abstains. The recoverable and systematic examples are illustrative.

Theorem 3 (Oracle greedy allocation)**.** *Starting with $n_i = 1$, repeatedly give two samples to the type with the largest available $G_{(n_i+1)/2}(\alpha_i, q_i)$ while the budget permits. This maximizes the stated objective and equals the exact dynamic-programming solution.*

Appendix A.3 gives an exchange-argument proof. If types have unequal weights and only the expected number of samples is budgeted, randomizing between adjacent odd counts makes a threshold rule on the marginal gain optimal. Without randomization, each type's two-sample step is an indivisible block whose cost depends on the type's weight, and the theorem does not apply. Unlike the oracle, CASC compares its scores for the current answer and for systematic error with fixed thresholds, so it neither estimates the per-query gains G_k nor inherits the optimality guarantee.

4.3 Correlation-Aware Self-Consistency (CASC)

CASC turns the population error model into a stopping rule for instances of unknown type (Figure 1). On labeled calibration data, it estimates the fraction of streams with no correct answer, the per-sample accuracy on the other streams, and how concentrated the wrong answers are (Appendix B). A target budget determines the stopping threshold. During inference, these fitted parameters stay fixed; CASC updates its scores after each observed answer and requires no correctness labels.

Streaming scores. CASC scores unanimous prefixes differently from votes split across several answers. A unanimous prefix may come from either SSM regime, but a second distinct answer rules out the systematic regime. Let c_a count answer a in the history h_n of the first n answers, let $\mathcal{O}_n$ be the set of distinct answers in h_n, and let $a^\star$ be the plurality leader, with ties broken by first appearance. Calibration supplies α, q and a *repeat weight* $r = (1-q)/\max(M_{\text{eff}} - 1, 10^{-6})$, a heuristic weight for each repeat of an already observed wrong answer. Here M_{eff} is the effective number of wrong answers, i.e., the number of equally frequent wrong answers that would give the observed concentration of errors (Appendix B). For positive q, r, the leader's score simplifies to

$$p_n = \begin{cases} \dfrac{(1-\alpha)q^n}{\alpha + (1-\alpha)q^n}, & |\mathcal{O}_n| = 1, \\ \dfrac{(q/r)^{c_{a^\star}}}{\sum_{a \in \mathcal{O}_n} (q/r)^{c_a}}, & |\mathcal{O}_n| > 1. \end{cases} \tag{7}$$

The systematic-error score is $\pi_n = 1 - p_n$ on unanimous prefixes and zero otherwise. In the first branch, more repetitions reduce p_n when $0 < \alpha < 1$ and $q < 1$, since longer perfect agreement looks more like a systematic error. In the second, the leader's score depends on its count relative to the other answers' counts, and it grows with that count only when $q > r$, which calibration does not

guarantee. Appendix B.2 derives these expressions from the likelihood approximation used in the implementation. They are heuristic; §5.4 tests how well they match observed correctness.

Decision rule. Before the cap K, at every odd $n \geq 3$, CASC:

- **Abstains** if $\pi_n \geq \gamma$, indicating high systematic-error risk.
- **Commits** to the leader otherwise if $p_n \geq 1 - \delta$.
- **Continues** otherwise, collecting another answer and updating the scores.

At K, CASC returns the current leader without another test. When methods are compared at a common budget B, δ is chosen on the calibration split to maximize accuracy subject to mean draws $\leq B$ (§5). The parameter γ controls optional abstention, which is turned off when every instance must be answered (full coverage). CASC does not retrain the answer-generating model and costs $O(K^2)$ per instance in the worst case; Algorithm 1 in Appendix C gives the pseudocode.

A rule that commits only when the exact posterior probability of $S = 0$ is below γ has a systematic-error rate of at most γ among committed answers (Proposition 3, Appendix A.4). This result does not cover CASC, which uses approximate scores and answers at the cap without the test.

5 Experiments

5.1 Tasks and comparison design

To measure voting limits directly, we train one-layer transformers on addition and multiplication modulo $p = 59$ (Power et al., 2022; Nanda et al., 2023).[2] All methods share the 1,392 training, 1,044 calibration and 1,045 evaluation pairs, checkpoints, temperature $T = 1$, and maximum of $K = 40$ draws. We train three seeds per task for 12,000 steps with AdamW and evaluate 14 checkpoints per run (Appendix D). All sampling policies use the same answer sequence for each instance. CASC is fitted on calibration labels; evaluation labels are used only for scoring.

Comparators and operating points. We compare SC at odd budgets from 1 to 39 and at 40 (SC@n uses n samples), ASC/Beta at $c \in \{.90, .95, .99\}$, an entropy-based stopping rule, and CASC over a fixed δ grid (Appendix D). The official ESC rule (Li et al., 2024) samples in windows of $w = 5$ answers by default. It stops when a window is unanimous and returns that answer; otherwise it returns the plurality of all draws at the cap. Greedy returns the most probable of the same 59 answers. It is a *white-box reference*, because it reads the full probability vector, which sampling policies do not see. Table 1 uses settings specified before evaluation and reports mean $\pm$ sample standard deviation over three training seeds.

For comparisons at a common budget B, each method picks its most accurate setting on the same calibration split among those with mean draws $\leq B$, breaking ties by lower cost. We use $B \in \{4, 8, 16, 24, 40\}$, ESC windows $\{1, 2, 4, 5, 8, 10, 20, 40\}$ and ASC/Beta thresholds $\{.75, .8, .85, .9, .925, .95, .975, .99, .999\}$. These grids include low-cost settings, and every ESC window divides the cap. Evaluation-split mean draws can differ from the calibration budget.

Cost accounting. Each answer is a single token, so one forward pass supplies the answer distribution and all 40 draws; Greedy uses one pass and zero draws. The number of consumed answers measures sample efficiency and is not proportional to latency; we time forward passes, sampling and stopping separately. CASC needs labeled calibration data; default ESC and ASC/Beta do not.

5.2 Sampling ceilings and allocation gains

SSM predicts that accuracy saturates and that allocating samples by query type helps (Appendix Figure 5). With $\alpha = .25, q = .70$, the curve approaches 0.75 and matches 200,000 Monte Carlo trials to within 1.2×10^{-3}. On 16 equally weighted types, oracle greedy matches the exact dynamic program (DP) at all 16 tested budgets. With six draws per query on average, it attains 57.42% accuracy; a type-blind allocation, which gives each query five or seven draws regardless of its type,

[2] Paired task values are ordered as addition / multiplication.

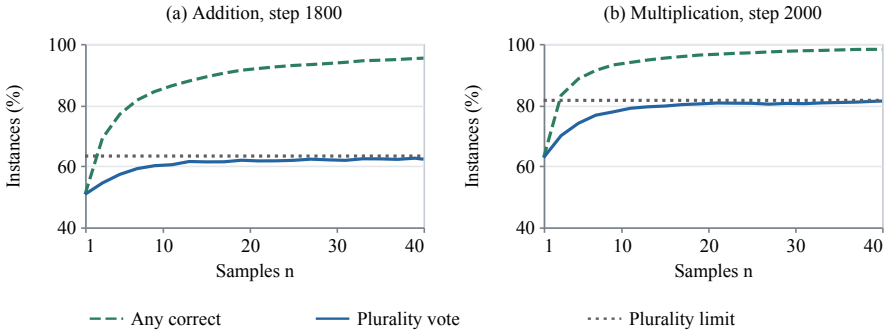


Figure 2: Availability exceeds voting accuracy at the transition checkpoints. Dashed: at least one correct answer in the first n draws; solid: plurality accuracy; dotted: exact plurality limit from each instance's mode. Curves average three seeds; Table 5 gives values and seed variation at $n = 40$.

Table 1: Paired policy comparison on identical checkpoints and answer streams. Entries are mean ± sample standard deviation over three training seeds; all methods answer every evaluation instance. Greedy sees the full distribution in one forward pass; zero draws does not mean zero computation. ESC uses the official $w = 5$ rule. Bold: best sampling-policy mean in each column.

	Addition @ 1800		Multiplication @ 2000	
Policy	Acc. (%) ↑	Draws ↓	Acc. (%) ↑	Draws ↓
Greedy (white-box)	63.3±36.7	0.0±0.0	81.5±26.0	0.0±0.0
Single sample	50.8±40.2	**1.0**±0.0	62.8±27.5	**1.0**±0.0
SC@5	57.3±38.9	5.0±0.0	74.0±28.9	5.0±0.0
SC@39	**62.6**±36.4	39.0±0.0	81.2±25.5	39.0±0.0
SC@40	62.3±36.7	40.0±0.0	**81.3**±25.5	40.0±0.0
ASC/Beta ($c = .95$)	62.4±36.6	18.5±12.1	81.1±25.8	14.1±9.2
Entropy ($h = .15$)	61.4±37.1	20.2±14.4	79.9±25.9	17.5±9.5
ESC ($w = 5$)	62.2±36.7	25.2±16.0	81.2±25.5	21.3±10.7
CASC ($\delta = .05$)	62.4±36.6	21.7±18.5	80.3±24.3	11.8±12.5

attains 56.56% at the same total cost. Appendix A.3 specifies the grid; Appendix E.7 gives the remaining comparisons with the model predictions.

5.3 Answer availability exceeds voting accuracy

Answer availability is the fraction of instances with at least one correct answer in the sampled stream; voting accuracy is the fraction whose plurality answer is correct. We report both at the *transition checkpoints*, addition step 1,800 and multiplication step 2,000, which fall within the grokking transition. There, availability at 40 draws is 95.34%/98.21%, yet SC@40 accuracy is only 62.30%/81.31%. The full answer probabilities explain this gap. Every evaluated instance at all 84 checkpoints has a unique mode, so the plurality limit equals Greedy accuracy, which is 63.32%/81.53% at these checkpoints (Appendix E.1). A correct answer can appear in the stream while additional votes reinforce a wrong mode.

Binary-majority limits are lower still, at 47.43%/69.28%, because 15.89%/12.25% of instances have a correct mode with probability at most $1/2$. On these instances the correct answer is more probable than any single wrong answer but not more probable than all wrong answers combined. Figure 2 shows the voting curves; Table 5 separates finite-vote *losses*, where the mode is correct but the vote is wrong, from *rescues*, where the mode is wrong but the vote is correct. Because the two partly cancel, aggregate accuracy can plateau before the votes on individual instances stop changing.

Appendix Figure 4 tracks how instances differ over training. At the transition checkpoints, instances fail SSM's third-moment test (Appendix B.1); the voting limits above do not use SSM.

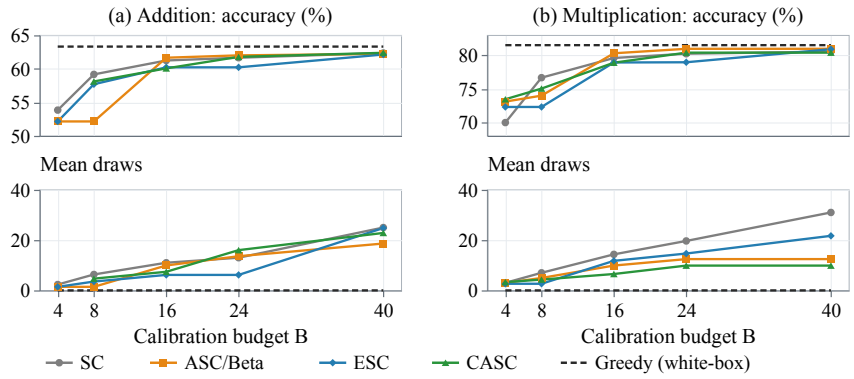


Figure 3: Comparison at common calibration budgets for addition (step 1800) and multiplication (step 2000). Top: evaluation accuracy; bottom: actual mean draws, both averaged over three seeds. Settings maximize calibration accuracy subject to mean draws $\leq B$. CASC has no feasible setting at addition $B = 4$. Greedy is a white-box reference.

5.4 Accuracy, sample cost and calibration

At the two transition checkpoints (Table 1), default CASC reduces mean draws relative to ESC from 25.23 to 21.66 on addition and from 21.29 to 11.76 on multiplication, with accuracy changes of $+0.19/-0.93$ points. ASC/Beta uses fewer draws than CASC at similar addition accuracy. Seeds grok at different steps, so results vary widely; Appendix E reports paired uncertainty.

Greedy decoding and tuned windows. Greedy reads off the unique mode directly, attaining $63.32\%/81.53\%$ in one forward pass. At addition step 12,000, ESC ($w = 2$), CASC and ASC/Beta all reach 100%, using 2.03, 3.00 and 4.07 draws. CASC's savings relative to default ASC/Beta therefore do not carry over to ESC with a tuned window.

Appendix Figure 6 relates accuracy to mean sample count. Figure 3 aligns methods at the same calibration budgets, showing evaluation accuracy and actual draw cost separately; Table 6 gives exact values. When the same setting is selected at successive budgets, the curves stay flat. On multiplication, CASC is both more accurate and cheaper on average than ASC/Beta at $B = 8$ (75.06% vs. 74.00%, 4.30 vs. 4.95 draws) and over ESC at $B = 24$ (80.41% vs. 78.98%, 9.86 vs. 14.66 draws). The ordering reverses on addition, where ESC is better on both means at $B = 16$, and ASC/Beta at $B = 24$. Three-seed means cannot rank methods statistically. The full grid has wins for each side and points where one method is more accurate and the other cheaper.

Effect of the systematic term. Holding all other parameters fixed, setting $\alpha = 0$ reduces default CASC cost from 21.66/11.76 to 19.71/10.89 draws, with accuracy differences of $-0.16/-0.10$ points. Paired intervals include zero. The term thus delays decisions on unanimous prefixes without a detectable accuracy benefit; Appendix E.3 adds budget-selected results.

Correctness calibration. Expected calibration error (ECE) groups predictions into confidence bins and averages the absolute gap between confidence and observed accuracy, weighting bins by size. It favors CASC for raw scores on addition (0.187 versus ASC/Beta's 0.275) but ASC/Beta on multiplication (0.115 versus 0.166). We then map each raw score to the observed accuracy of its bin, using the same ten fixed bins for both methods, fitted only on calibration labels. This correctness map reduces ASC/Beta's ECE to 0.011/0.026 and CASC's to 0.014/0.030. The maps change neither answers nor draw counts. Mapped Brier scores (mean squared confidence error) are 0.139/0.103 for ASC/Beta and 0.141/0.115 for CASC, and three seeds do not resolve the paired differences (Appendix E.4). After both scores are mapped to observed accuracy, CASC no longer has the calibration advantage that its raw scores show on addition.

Selective prediction. Allowing CASC to abstain tests whether it can identify errors worth rejecting. On arithmetic, abstention thresholds selected on calibration data keep $99.01\%/100\%$ coverage

(the fraction answered) and show no detectable advantage over other selectors that use the same labels. The large gain from abstention on synthetic data does not carry over (Appendix E.5).

5.5 Stopping on public language-model responses

For reasoning tasks, we compare stopping rules on ESC's public GSM8K answers from GPT-3.5, GPT-4 and Llama-2 7B (Li et al., 2024; Cobbe et al., 2021). This is an offline comparison with no model API calls. The 1,318 questions shared by all three banks have 40 answers each; we split them into 659 calibration and 659 evaluation questions. CASC and the official ASC/Beta and ESC implementations use the same ESC answer parser and the same five random orders of each question's answers. Appendix F gives sources, selection rules and paired bootstrap intervals over questions.

Table 2 shows that the tradeoff changes by model. On GPT-4, default CASC uses 3.67 answers versus ESC's 8.24, with accuracy 89.83% versus 89.44%. A paired bootstrap over questions gives a 95% confidence interval (CI) of $[-0.18, 1.00]$ percentage points for the accuracy difference, which includes zero. On GPT-3.5, ASC/Beta is more accurate and cheaper than CASC. Removing CASC's systematic term reduces GPT-3.5 cost from 20.40 to 5.12 answers for a -0.03-point accuracy change; GPT-4 outputs are unchanged. Here, too, the term mainly delays stopping, by an amount that depends on the model, without a detectable accuracy benefit at default settings. Selecting settings for each budget changes accuracy and cost, but no method is best on every model (Table 11).

Table 2: Default stopping policies on ESC's public GSM8K answer banks: 659 evaluation questions, five shared answer orders, every question answered. Draws count answers consumed from the public banks. Bold marks column-best point estimates; paired intervals are in Table 12.

	GPT-3.5		GPT-4		Llama-2 7B	
Method	Acc. (%)	Draws	Acc. (%)	Draws	Acc. (%)	Draws
SC@40	**85.92**	40.00	89.56	40.00	19.51	40.00
ASC/Beta ($c = .95$)	85.86	10.13	89.65	6.16	19.45	24.11
ESC ($w = 5$)	85.74	15.37	89.44	8.24	**19.54**	31.28
CASC ($\delta = .05$)	85.13	20.40	**89.83**	**3.67**	19.48	25.24
CASC, $\alpha = 0$	85.10	**5.12**	**89.83**	**3.67**	19.48	**21.24**

6 Discussion and Limitations

When another sample helps. Low confidence does not by itself make more samples worthwhile, because a wrong mode can persist as the vote stabilizes. CASC's systematic-error term delays stopping on unanimous prefixes but rarely changes their final answers. The oracle instead values extra draws by how often they correct a wrong vote minus how often they overturn a correct one.

Scope of the evidence. Arithmetic lets us inspect exact answer probabilities; GSM8K adds multi-step text reasoning, but from one dataset and three previously released response banks. The exact model versions behind these responses are unknown, the evaluation reuses fixed answers, and calibration may not transfer to new data. These factors limit conclusions about deployment or latency.

Scope of the model. SSM's *two-point correctness law*, under which each instance's per-sample accuracy is either zero or q, does not exactly fit the learned populations. A correct mode can have probability below $1/2$, so the diminishing binary-majority gains and the oracle guarantee do not extend to arbitrary plurality distributions.

7 Conclusion

When the model has a unique most probable answer, repeated voting converges to it, even when it is wrong. Under SSM, each extra pair of samples gains less than the last, so greedy oracle allocation is optimal. CASC's systematic-error term mainly delays decisions, and mapping scores to observed accuracy improves calibration without changing those decisions. Future stopping rules should therefore estimate the gain from more samples separately from confidence in the current answer.

AI USE STATEMENT

AI tools assisted with drafting and polishing the manuscript and with retrieving and discovering related literature. The authors reviewed and verified all resulting text and references and take responsibility for the final manuscript.

ETHICS STATEMENT

The experiments use procedurally generated arithmetic tasks and publicly released GSM8K response banks, with no new human-participant study. The main risk is to take stable model agreement as evidence of correctness. Our results do not show that such stopping rules are safe to deploy or that they reduce latency proportionally in language models.

REPRODUCIBILITY STATEMENT

Section 5 and Appendix D specify the model, data partitions, seeds, shared sample streams and official ESC adaptation. Appendix F specifies the GSM8K data versions, official baseline sources, shared data split and question-level uncertainty. Appendix E reports the budget comparisons and uncertainty. The associated code and records retain model checkpoints, per-instance predictions, stopping times, selected configurations and dependency versions needed to recompute the reported comparisons.

REFERENCES

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

Yong Yi Bay and Kathleen A. Yearick. When More Sampling Hurts: The Modal Ceiling and Correlation Ceiling of Test-Time Scaling, 2026. URL https://arxiv.org/abs/2606.28661. arXiv:2606.28661.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-Thought: Scaling Test-Time Compute for Enhancing LLM Reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4253–4267. PMLR, 2025. URL https://proceedings.mlr.press/v267/bi25a.html.

Ahsan Bilal, Muhammad Ahmed Mohsin, Muhammad Umer, Lena Trigg, Ali Subhan, Muhammad Ali, and Dean F. Hougen. Refining Over Resampling: Test-Time Self-Correction for LLM Reasoning, 2026. URL https://arxiv.org/abs/2608.05643. arXiv:2608.05643.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling, 2024. URL https://arxiv.org/abs/2407.21787. arXiv:2407.21787.

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. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168. arXiv:2110.14168.

Paula Cordero-Encinar and Andrew B. Duncan. Certified Self-Consistency: Statistical Guarantees and Test-Time Training for Reliable Reasoning in LLMs, 2025. URL https://arxiv.org/abs/2510.17472. arXiv preprint 2510.17472.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. arXiv:2501.12948.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. BEST-Route: Adaptive LLM Routing with Test-Time Optimal Compute. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13870–13884. PMLR, 2025. URL https://proceedings.mlr.press/v267/ding25d.html.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-AOS1991. URL https://arxiv.org/abs/1810.08240.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is Best-of-N the Best of Them? Coverage, Scaling, and Optimality in Inference-Time Alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25075–25126. PMLR, 2025a. URL https://proceedings.mlr.press/v267/huang25c.html.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. Efficient Test-Time Scaling via Self-Calibration, 2025b. URL https://arxiv.org/abs/2503.00031. arXiv:2503.00031.

William Jurayj, Jeffrey Cheng, and Benjamin Van Durme. Is That Your Final Answer? Test-Time Scaling Improves Selective Question Answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 636–644. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-short.50. URL https://aclanthology.org/2025.acl-short.50/.

Junseok Kim, Nakyeong Yang, Kyungmin Min, and Kyomin Jung. Reliability-Aware Adaptive Self-Consistency for Efficient Sampling in LLM Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21575–21590. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-acl.1085. URL https://aclanthology.org/2026.findings-acl.1085/.

Junyeob Kim, Sang-goo Lee, and Taeuk Kim. Think Just Enough: Leveraging Self-Assessed Confidence for Adaptive Reasoning in Language Models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 5000–5006. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-eacl.263. URL https://aclanthology.org/2026.findings-eacl.263/.

Jaehyeok Lee, Keisuke Sakaguchi, and JinYeong Bak. Self-Training Meets Consistency: Improving LLMs' Reasoning with Consistency-Driven Rationale Evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10519–10539. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.528. URL https://aclanthology.org/2025.naacl-long.528/.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape Sky-high Cost: Early-stopping Self-Consistency for Multi-step Reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. doi: 10.48550/arxiv.2401.10480. URL https://arxiv.org/abs/2401.10480.

Zhangyi Liu, Huaizhi Qu, Xiaowei Yin, He Sun, Yanjun Han, Tianlong Chen, and Zhun Deng. PETS: A Principled Framework Towards Optimal Trajectory Allocation for Efficient Test-Time Self-Consistency, 2026. URL https://arxiv.org/abs/2602.16745. arXiv:2602.16745.

Ziming Liu, Ouail Kitouni, Niklas S. Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *NeurIPS*, 2022. arXiv:2205.10343.

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

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *ICLR*, 2023. arXiv:2301.05217.

Hirofumi Ota, Naoto Iwase, Yuki Ichihara, Junpei Komiyama, and Masaaki Imaizumi. CITE: Anytime-Valid Statistical Inference in LLM Self-Consistency, 2026. URL https://arxiv.org/abs/2605.05873. arXiv preprint 2605.05873.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedavyas Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. arXiv:2201.02177.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing Test-Time Compute via Meta Reinforcement Finetuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50893–50925. PMLR, 2025. URL https://proceedings.mlr.press/v267/qu25g.html.

Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How Do Large Language Monkeys Get Their Power (Laws)?, 2025. URL https://arxiv.org/abs/2502.17578. arXiv:2502.17578.

Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. Scaling Test-Time Compute Without Verification or RL is Suboptimal. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54058–54094. PMLR, 2025. URL https://proceedings.mlr.press/v267/setlur25a.html.

Aman Sharma and Paras Chopra. Think Just Enough: Sequence-Level Entropy as a Confidence Signal for LLM Reasoning, 2025. URL https://arxiv.org/abs/2510.08146. arXiv:2510.08146.

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

Renliang Sun, Wei Cheng, Dawei Li, Haifeng Chen, and Wei Wang. Stop When Enough: Adaptive Early-Stopping for Chain-of-Thought Reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27250–27268. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1256. URL https://aclanthology.org/2026.acl-long.1256/.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Remi Munos. Optimizing Language Models for Inference Time Objectives using Reinforcement Learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59066–59085. PMLR, 2025. URL https://proceedings.mlr.press/v267/tang25o.html.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence Improves Self-Consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20090–20111. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1030. URL https://aclanthology.org/2025.findings-acl.1030/.

Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Lifeng Jin, Haitao Mi, Jinsong Su, and Dong Yu. Self-Consistency Boosts Calibration for Math Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6023–6029. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.349. URL https://aclanthology.org/2024.findings-emnlp.349/.

Tianchun Wang, Yuanzhou Chen, Zichuan Liu, Jonathan Light, Weiyang Liu, Haifeng Chen, Xiang Zhang, and Wei Cheng. On the Effect of Sampling Diversity in Scaling LLM Inference. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 7137–7167. PMLR, 2026. URL https://proceedings.mlr.press/v337/wang26f.html.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6919–6932, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.383. URL https://aclanthology.org/2025.findings-naacl.383/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022. arXiv:2201.11903.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable Chain-of-Thought Compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.165. URL https://aclanthology.org/2025.emnlp-main.165/.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?, 2025. URL https://arxiv.org/abs/2504.13837. arXiv:2504.13837.

Qingcheng Zeng, Weihao Xuan, Leyang Cui, and Rob Voigt. Thinking Out Loud: Do Reasoning Models Know When They're Right? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1394–1407. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.73. URL https://aclanthology.org/2025.emnlp-main.73/.

Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. LightThinker: Thinking Step-by-Step Compression. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13307–13328. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.673. URL https://aclanthology.org/2025.emnlp-main.673/.

Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. Sample, Scrutinize and Scale: Effective Inference-Time Search by Scaling Verification. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 77272–77309. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhao25a.html.

A Proofs and Derivations

Pairwise correlation (Eq. 2). $p = (1-\alpha)q$ and $\mathbb{E}[X_t X_u] = (1-\alpha)q^2$, so $\mathrm{Cov} = \alpha(1-\alpha)q^2$ and $\mathrm{Corr} = \alpha q/[1-(1-\alpha)q]$.

Variance and n_{eff} (Eq. 3). By exchangeability, $\mathrm{Var}(\sum_t X_t) = nv + n(n-1)c$ with $v = p(1-p)$ and $c = \rho v$; dividing by n^2 and equating with v/n_{eff} gives the stated expression, tending to $1/\rho$.

Identifiability (Proposition 1). Eq. 2 gives $\rho(1-p) = \alpha q$ while $p = (1-\alpha)q$, hence $\rho(1-p)/p = \alpha/(1-\alpha)$ and the stated formulas.

Beta confidence on systematic instances (Proposition 2). With $v_1 = n, v_2 = 0$ and posterior $\mathrm{Beta}(n+1, 1)$, $I_{1/2}(1, n+1) = (n+1)\int_0^{1/2}(1-t)^n\,dt = 1 - 2^{-(n+1)}$; solving $c_n \geq c_{\text{thr}}$ gives $n^\star = \lceil \log_2[1/(1-c_{\text{thr}})] \rceil - 1$.

A.1 Proof of Theorem 1

Proof. Conditional on $S = 1$, the correct count is $C \sim \mathrm{Bin}(n, q)$, giving $B(n, q)$ for binary voting with fair ties. On $S = 0$ every answer is wrong. The mixture therefore gives Eq. 1. Hoeffding bounds $1 - B(n, q)$ by $\Pr(C \leq n/2) \leq e^{-2n(q-1/2)^2}$. For odd n, a strict correct majority also wins categorical plurality, proving the lower bound. Since $q > 1/2$ exceeds each wrong-category probability, the law of large numbers gives plurality success tending to one in the recoverable regime and zero in the systematic regime. For binary random effects, the pointwise limits are one, zero and one-half at $q_i > 1/2$, $q_i < 1/2$ and $q_i = 1/2$, respectively; bounded convergence permits averaging. □

A.2 Proof of Theorem 2

Proof. Fair ties imply $B(2k, q) = B(2k-1, q)$ by conditioning on the final vote. Adding one vote to $2k$ trials changes success only when the first $2k$ are tied, replacing tie success $1/2$ with success q. Thus $B(2k+1, q) - B(2k-1, q) = \binom{2k}{k}[q(1-q)]^k(q - 1/2)$. Multiplying by $1-\alpha$ gives Eq. 5; the ratio of consecutive central binomial coefficients is $2(2k+1)/(k+1)$, proving diminishing returns. Differentiating $\log[(q-1/2)[q(1-q)]^k]$ sets $q(1-q) = 2k(q-1/2)^2$, yielding $q_k^\star$. The gain is positive inside $(1/2, 1)$ and zero at either endpoint, so this unique stationary point is the maximum. □

A.3 Proof of Theorem 3

Proof. There are $L = \lfloor (T-N)/2 \rfloor$ available sample pairs after the initial N draws. Each type has a nonincreasing sequence $G_1, G_2, \ldots$. Selecting a pair from that type requires selecting its predecessors. At every step, the largest available marginal gain is also a largest unselected gain overall, because every gain that is not yet available has an available predecessor at least as large. Greedy therefore selects the L largest gains, or all gains if the cap is reached. This selection is feasible and no other L pairs have greater sum. The exact DP obeys $V_i(t) = \max_{n \in \{1,3,\ldots\},\, n \leq t}\{V_{i-1}(t-n) + A(n; \alpha_i, q_i)/N\}$, with $V_0(t) = 0$ for $t \geq 0$ and infeasible states equal to $-\infty$; it optimizes the same feasible set. □

For a finite distribution of types with weights w_i, adding two samples to type i costs $2w_i$ in expectation and gains $w_i G_k$. If each type may randomize its sample count, these increments become divisible: take them in decreasing order of $G_k/2$ and randomize at the final threshold if necessary. Without randomization, each type's increment is an indivisible block, and this greedy solution need not be optimal.

Numerical allocation example. The grid has 16 equally weighted types: $\alpha \in \{.05, .20, .40, .60\}$ and $q \in \{.55, .65, .75, .90\}$, with odd counts from 1 to 31. We test mean budgets $B = 1, \ldots, 16$, equivalent to total budgets $T = 16B$. At $B = 6$, greedy matches the DP optimum (0.5742), versus 0.5656 for balanced type-blind allocation at the same cost of 96 draws (defined in Fig. 5). Uniform

five-draw sampling achieves 0.5579 at cost 80; mean-matched random allocation achieves 0.5553. At $\alpha = .05$, the oracle assigns five draws to $q = .55$ but 15 to $q = .65$: extra draws improve the more uncertain type less, so it receives fewer. Every type has diminishing gains (Fig. 5, bottom left).

A.4 Ideal posterior gating

Proposition 3 (Ideal posterior gating). *Let $\pi_n^{\mathrm{ex}} = \Pr(S = 0 \mid h_n)$ be an exact posterior under a correctly specified model. If an adaptive rule commits only on histories satisfying $\pi_n^{\mathrm{ex}} < \gamma$, then, whenever it commits with positive probability, its committed systematic-error rate is at most γ.*

This follows by conditioning on the stopping history and averaging the posterior bound. The bound covers only the systematic component of error. We use it only as a reference point for CASC, because Eq. 8 is approximate and CASC answers at K without applying the gate. The proposition therefore gives no guarantee for the full set of answers that CASC commits to.

B Estimation Details

Labeled calibration. For N_c labeled streams of length K, let z_i be the correct-answer count and $\hat{p} = N_c^{-1} \sum_i z_i/K$. CASC estimates the fraction of streams with no correct answer, $\hat{\alpha}_K = N_c^{-1} \sum_i \mathbf{1}\{z_i = 0\}$ (the *zero-count estimator*), and $\hat{q} = \sum_{i:z_i>0} z_i/(K \sum_i \mathbf{1}\{z_i > 0\})$. Its distinct-pair estimate is

$$\widehat{m}_2 = \frac{1}{N_c} \sum_i \frac{z_i(z_i - 1)}{K(K-1)}, \qquad \hat{\rho} = \max\left\{0, \frac{\widehat{m}_2 - \hat{p}^2}{\hat{p}(1-\hat{p}) + 10^{-12}}\right\}.$$

The pair correction excludes self-pairs; using $(z_i/K)^2$ would include finite-stream sampling variance. Proposition 1 concerns the population moments, whereas CASC uses the zero-count estimator above.

Under the exact SSM, $\mathbb{E}\hat{\alpha}_K = \alpha + (1-\alpha)(1-q)^K$ and the mean correctness conditional on at least one success is $q/[1-(1-q)^K]$. With heterogeneous q_i, the zero-count fraction estimates $\mathbb{E}[(1-q_i)^K]$, not $\Pr(q_i = 0)$. In particular, a softmax with finite logits gives every answer positive probability, so in the real testbed the systematic state (always returning the lure) only approximates instances whose correct answer is very unlikely.

For each instance with at least two wrong draws, normalize the wrong-answer counts to w_{ia} and compute $(\sum_a w_{ia}^2)^{-1}$; their mean is M_{eff}, the effective number of wrong answers. This estimate determines the repeat weight in Eq. 8. Because M_{eff} already counts only wrong answers, the $M_{\mathrm{eff}} - 1$ in the denominator of r is not the number of equally likely wrong options; r is a heuristic weight and can exceed one.

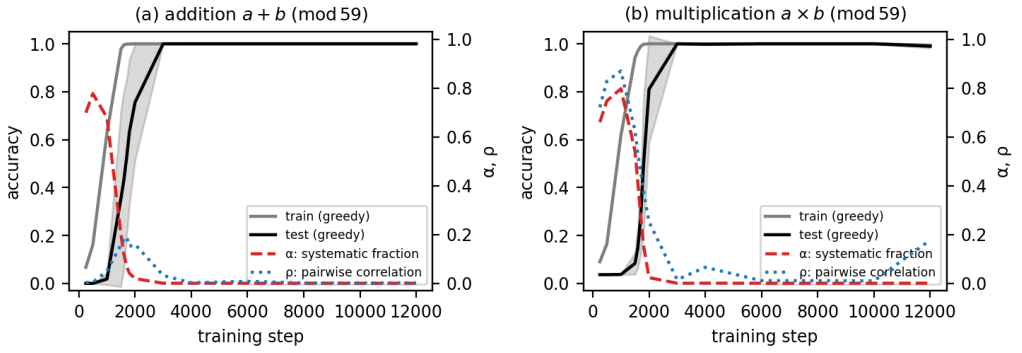


Figure 4: Greedy train/test accuracy, zero-correct-stream fraction $\hat{\alpha}_K$ (dashed, labeled α), and pairwise correlation $\hat{\rho}$ (dotted) along training, mean over three seeds (shaded: test std). The zero-correct-sample fraction declines with learning; residual correlation remains in multiplication.

B.1 TESTING THE TWO-POINT CORRECTNESS LAW

Identifiability within SSM does not imply that a population follows SSM. Let $Q = P_x(y^\star) \in [0,1]$ and $m_j = \mathbb{E}[Q^j]$ for $j = 1,2,3$. For conditionally independent samples, m_j is also the probability that j distinct draws on a randomly selected instance are all correct. SSM imposes $m_j = (1-\alpha)q^j$, hence a testable third-moment restriction.

Proposition 4 (Correctness-moment diagnostic)**.** *For any distribution of $Q \in [0,1]$ with $m_1 > 0$, define*

$$D = 1 - \frac{m_2^2}{m_1 m_3}, \qquad q_{\mathrm{m}} = \frac{m_2}{m_1}, \qquad \alpha_{\mathrm{m}} = 1 - \frac{m_1^2}{m_2}.$$

Then $D \geq 0$, with equality exactly when Q takes a single positive value almost surely on $\{Q > 0\}$. The pair $(\alpha_{\mathrm{m}}, q_{\mathrm{m}})$ always matches the first two moments. The original distribution of Q has the SSM correctness law if and only if $D = 0$ and $1/2 < q_{\mathrm{m}} < 1$.

Proof. Cauchy–Schwarz applied to $Q^{1/2}$ and $Q^{3/2}$ gives $m_2^2 \leq m_1 m_3$. Equality requires $Q^{3/2} = cQ^{1/2}$ almost surely, so every positive Q equals c. Conversely, a law on $\{0, c\}$ attains equality. Substituting q_{m} and α_{m} recovers m_1, m_2; $m_1^2 \leq m_2 \leq m_1$ ensures valid mixture weights. Equivalently, under the probability measure weighting instances by Q/m_1, $D = \mathrm{Var}_Q(Q)/\mathbb{E}_Q[Q^2]$. □

Even $D = 0$ only validates the two-point correctness law: wrong categories at $Q = 0$ could still vary, whereas SSM additionally requires one fixed lure. The diagnostic tests a necessary part of SSM without claiming to test all its categorical assumptions.

At the transition checkpoints, matching two moments does not explain how correctness varies across instances (Table 3). The moments use the full normalized answer probabilities on the evaluation split, so the positive values of D are not due to finite-stream estimation noise. The moment-implied systematic fraction is $24.18\%/16.29\%$, far above even the expected zero-correct 40-draw fractions $\alpha_{40} = \mathbb{E}[(1-Q)^{40}]$ of $4.28\%/1.92\%$. Every correct-answer probability is positive at these checkpoints. Thus α_{m} is an effective moment parameter here, not a count of instances that can never produce a correct answer. CASC uses the zero-count estimator, not this moment-implied pair; this analysis changes neither its parameters nor its decisions.

Table 3: SSM correctness-law diagnostic from model answer probabilities. Entries are three-seed mean ± sample standard deviation, expressed in percent. Exact SSM requires $D = 0$; positive D measures variation among positive correctness probabilities. The moment pair is descriptive and is not used to refit CASC.

Quantity (%)	Addition @ 1800	Multiplication @ 2000
Third-moment discrepancy D	12.56±11.96	8.19±9.40
Moment-implied q_{m}	61.35±30.55	73.92±18.20
Moment-implied α_{m}	24.18±23.62	16.29±19.24
Expected zero-correct fraction α_{40}	4.28±6.06	1.92±3.24

Across the 84 evaluated model states, D ranges from 1.0×10^{-8} to 0.646, approaching zero for nearly constant correct-answer probabilities. At 22 checkpoints, $q_{\mathrm{m}} \leq 1/2$ also violates the SSM parameter range. These are repeated observations of six training runs rather than 84 independent tests. The diagnostic neither bounds the prediction error of voting nor attributes any particular CASC decision to model mismatch.

B.2 WHAT THE STREAMING SCORES MEASURE

CASC's scores are motivated by the SSM likelihood. Besides the repeat weight r defined in §4.3, they use a *novelty weight*, which scores the first appearance of each additional distinct answer, $\varepsilon = \max\{(1-q-r)/\max(M-2,1), 10^{-6}\}$. With $d = |\mathcal{O}_n|$, its unnormalized likelihood terms

are

$$\widetilde{L}_0(h_n) = M^{-1}\mathbf{1}\{d = 1\},$$
$$\widetilde{L}_1(h_n) = M^{-1} \sum_{a \in \mathcal{O}_n} q^{c_a} r^{n-c_a-(d-1)} \varepsilon^{d-1}, \tag{8}$$
$$\pi_n = \frac{\alpha \widetilde{L}_0}{\alpha \widetilde{L}_0 + (1-\alpha)\widetilde{L}_1}.$$

Writing $t_a = q^{c_a} r^{n-c_a-(d-1)} \varepsilon^{d-1}$, the leader receives support $\eta_n = t_{a^\star} / \sum_{a \in \mathcal{O}_n} t_a$ and $p_n = (1 - \pi_n)\eta_n$. Unobserved candidates are omitted and the weights need not define normalized categorical likelihoods.

For positive q, r, ε, each support term factors as

$$t_a = r^{n-d+1} \varepsilon^{d-1} (q/r)^{c_a}.$$

The common factor cancels from η_n. Combining this with $\widetilde{L}_0 = 0$ whenever $d > 1$ gives Eq. 7. Thus the novelty weight cancels from every decision in exact arithmetic. Once two answers have appeared, the systematic score remains zero, even if the eventual mode is wrong. At $q = r$ the support score is uniform over observed categories, and at $q < r$ it favors lower counts. Because calibration does not enforce the theoretical SSM range $q > 1/2$, the last case can arise on real streams.

This reduction makes an $\alpha = 0$ ablation well defined. Holding q, r and every stopping setting fixed changes only the scores of unanimous prefixes, which become one. At full coverage and fixed $\delta \in (0, 1)$, this variant cannot stop later than CASC on the same stream; any accuracy change is due to accepting an earlier unanimous prefix. Appendix E.3 measures that tradeoff without refitting the other parameters.

Exact symmetric posterior. With a uniform prior on the correct category and uniform wrong-answer mass $r_0 = (1 - q)/(M - 1)$, the normalized recoverable likelihood is $L_1^{\mathrm{ex}}(h_n) = M^{-1} \sum_{a=1}^{M} q^{c_a} r_0^{n-c_a}$, including every unobserved category with $c_a = 0$. This specifies the model used for the unanimous-stream odds in §3.3. CASC instead uses Eq. 8. A labeled calibration set is used throughout, and we make no claim about identifying correctness without labels.

C CASC Pseudocode

Algorithm 1: CASC per instance; population parameters $\alpha, q, r, \varepsilon, M$, thresholds δ, γ, cap K

$v \leftarrow$ empty vote counts
for $t = 1, \ldots, K$:
  draw Y_t; $v[Y_t]$ += 1; $n \leftarrow t$
  if $n = K$: **return** leader
  if $n < 3$ or n even: continue
  compute $\widetilde{L}_0, \widetilde{L}_1, \pi_n, \eta_n, p_n$ (Eq. 8)
  if $\pi_n \geq \gamma$: **return** ABSTAIN
  if $p_n \geq 1 - \delta$: **return** leader
return leader (cap)

Per-step cost is $O(|\mathrm{supp}\, v|) \leq O(n)$, hence $O(K^2)$ worst case and $O(K)$ amortized per draw; memory $O(\min\{M, K\})$ for an online count map (the offline evaluator also stores trajectories). For budget-conditioned comparisons, select δ on calibration streams to maximize accuracy with mean draws $\leq B$, breaking ties by lower cost. The parameter γ controls optional abstention via the approximate systematic score; full-coverage comparisons disable that action. Proposition 3 applies only to exact posterior gates.

D Experimental Details

Architecture. Token and learned positional embeddings, one pre-norm Transformer encoder layer ($d = 64$, 4 heads, feed-forward width 128, Gaussian error linear unit activation), layer norm, zero-initialized linear answer head, 41,472 parameters. **Optimization.** AdamW ($\beta = (.9, .98)$, lr 10^{-3},

weight decay 1.0), batch 256, 12,000 steps; checkpoints at 250, 500, 1000, 1500, 1600, 1700, 1800, 2000, 3000, 4000, 6000, 8000, 10000 and 12000; 40% train split (seed 1234). Training seeds are $0, 1, 2$. The held-out split is permuted with NumPy generator seed 7, with 1,044 calibration and 1,045 evaluation instances. **Sampling.** One forward pass yields answer logits over the same 59 legal categories; softmax at $T = 1$; 40 independent multinomial draws. A separate generator is seeded by $10^6 + 10^5\mathbf{1}\{\text{mul}\} + 20000s + t$ for training seed s and checkpoint t. All policies use these draws in the same order. **Runtime.** Python 3.12.14, PyTorch 2.8.0, NumPy 2.2.6, SciPy 1.15.3, macOS arm64, CPU with two PyTorch threads. The six training runs take 27.4 to 34.1 seconds each on this CPU. The code includes dependencies and results for each evaluated model state.

ESC decision rule and implementation. The official ESC rule receives answer strings in the shared evaluator's stream order.[3] Only the benchmark data-loading code is replaced, preserving unanimous-window return and cumulative plurality at the cap. The resulting decisions agree with an independent implementation on 23,267 exhaustive or randomized test sequences. All reported ESC windows divide the cap, avoiding changes to the official treatment of incomplete windows.

Shared evaluation details. SC, ASC/Beta, entropy and CASC break modal ties by first appearance among the tied leaders. The entropy rule begins checking at two samples; it stops when the empirical answer entropy divided by $\log n$ is at most h, or returns the leader at the cap. CASC's no-abstention comparison uses $\gamma = 1.01$, and its delta grid is $\{.7, .6, .5, .3, .2, .1, .05, .02, .01\}$. Default settings and calibration-selected settings are reported separately. ESC's original benchmark scores are not transferred to these different models or datasets.

E Additional Results

This appendix reports the voting limit, stopping cost, correctness calibration and selective prediction separately. They cover all 14 checkpoints per task and all three seeds. Per-seed accuracy, sample count, coverage and agreement with Greedy are available with the per-instance predictions and stopping times.

Uncertainty across training seeds. At the preselected transition checkpoints, CASC minus default ESC accuracy is $+0.19$ percentage points on addition and -0.93 on multiplication. Using the three paired seed differences as the uncertainty units, two-sided 95% Student-t intervals are $[-1.07, 1.45]$ and $[-4.03, 2.18]$ points. The intervals are wide, and neither method has a clear accuracy advantage. We do not treat repeated checkpoints or individual samples as independent training runs.

Sample cost at high accuracy. At step 12,000 on addition, Greedy, ASC/Beta, ESC with $w = 2$, ESC with $w = 5$ and CASC all attain 100% accuracy. Their mean categorical draw counts are 0, 4.07, 2.03, 5.17 and 3.00, respectively. Multiplication remains less uniform: Greedy, ASC/Beta, ESC with $w = 5$ and CASC attain 99.20%, 99.07%, 99.11% and 98.76%, respectively. We therefore assume neither complete grokking nor identical accuracy when interpreting a reduction in samples.

Measured cost components. For a batch of all 2,089 held-out instances, the mean of the median forward-pass times at the two transition checkpoints is 4.87 ms (addition) and 4.46 ms (multiplication); generating all 40 draws costs 1.42 and 0.88 ms. Policy timings measure decisions on previously generated answer streams. CASC's offline evaluator computes its full trajectory before testing thresholds; those timings are not measurements of an optimized online sampler. We therefore make no end-to-end latency claim from these replay experiments.

E.1 Identifying the categorical sampling limit

Let $P_x(a)$ denote the probability of answer a at a fixed instance x, and let V_n be the plurality of n conditionally i.i.d. draws, with any fixed rule for vote ties. This setting allows arbitrary instance heterogeneity and does not assume the SSM.

[3] We use `early_stop_gsm8k.py` from the official ESC repository, commit `27574d4`; the code manifest records the full revision and SHA-256.

Table 4: Calibration-selected settings at two budgets. Entries are evaluation mean ± sample standard deviation over three seeds. A method is marked infeasible when its prespecified grid contains no setting within the calibration budget for at least one seed. Draws are realized evaluation costs; threshold choices use calibration labels only. Bold marks the highest accuracy and fewest draws within each task and budget.

Policy	$B = 16$		$B = 24$	
	Acc. (%)	Draws	Acc. (%)	Draws
Addition				
SC	61.2±37.2	11.0±5.3	61.7±37.2	13.0±8.0
ASC/Beta	**61.7**±37.0	9.9±6.8	**62.0**±37.0	13.6±10.6
ESC	60.2±37.3	**6.2**±3.5	60.2±37.3	**6.2**±3.5
CASC	60.1±40.0	7.4±5.6	61.8±37.8	16.0±11.0
Multiplication				
SC	79.6±26.3	14.3±1.2	80.3±26.9	19.7±4.2
ASC/Beta	**80.3**±26.7	9.9±3.7	**81.0**±25.6	12.4±7.9
ESC	78.9±29.2	11.8±2.8	79.0±29.2	14.7±7.4
CASC	78.9±27.2	**6.5**±2.7	80.4±24.6	**9.9**±8.5

Proposition 5 (Plurality limit and finite-vote decomposition)**.** *Suppose the answer set is finite and P_x has a unique mode $m(x)$ for almost every instance. Then*

$$A_\infty^{\mathrm{pl}} := \lim_{n\to\infty} \Pr(V_n = y^\star) = \Pr(m(x) = y^\star).$$

For every finite n, define $L_n = \Pr(m(x) = y^\star, V_n \neq y^\star)$ and $R_n = \Pr(m(x) \neq y^\star, V_n = y^\star)$. Then

$$\Pr(V_n = y^\star) = A_\infty^{\mathrm{pl}} - L_n + R_n.$$

Proof. For each instance with a unique mode, the finitely many empirical category frequencies converge almost surely to P_x. Because the mode's probability exceeds every other answer's by a positive gap, $V_n = m(x)$ for all large n. Bounded convergence gives the population limit. Partitioning correct finite votes according to whether the mode is correct gives the decomposition. For example, a binary instance with correct-answer probability .4 has limit zero but one-draw accuracy .4. Thus the limit need not be a finite-sample upper bound. □

The decomposition exposes two opposing sources of finite-vote error, with no stopping policy to tune. All $84 \times 1{,}045$ evaluated combinations of instances and model states have unique probability modes. SC@40 uses first-appearance ties and satisfies the decomposition for every checkpoint and seed. Binary-majority limits use the true correct-category probabilities $q_i = P_{x_i}(y_i^\star)$, not their noisy empirical frequencies. Table 5 reports the same two transition checkpoints as Table 1; uncertainty uses the three training seeds, not repeated checkpoints.

Table 5: Separating answer availability, asymptotic voting and finite-vote errors. Entries are mean ± sample standard deviation over three seeds, in percent of all evaluation instances. The plurality limit equals Greedy accuracy; SC@40 equals that limit minus loss plus rescue in every seed. Answer availability (at least one correct draw) is substantially higher than voting accuracy.

Quantity (%)	Addition @ 1800	Multiplication @ 2000
Ever correct in 40 draws	95.34±6.86	98.21±3.01
Binary-majority limit	47.43±46.81	69.28±38.11
Plurality limit A_∞^{pl}	63.32±36.65	81.53±25.99
SC@40 accuracy	62.30±36.73	81.31±25.54
Finite-vote loss L_{40}	3.06±2.76	1.98±1.77
Finite-vote rescue R_{40}	2.04±1.79	1.75±2.23

Finite-vote loss and rescue average 3.06/2.04 percentage points on addition and 1.98/1.75 on multiplication. The small net differences hide considerable turnover in which instances are answered

correctly, so a plateau in population accuracy does not mean that each instance's vote has converged. The expected fractions of instances with at least one correct draw, $N_e^{-1}\sum_i[1-(1-q_i)^{40}]$, are $95.72\%/98.08\%$, consistent with the observed $95.34\%/98.21\%$; neither is an asymptotic plurality ceiling. The released results give this decomposition at every checkpoint.

For context, the calibration zero-correct fraction starts at $0.70/0.66$ at step 250, and estimated correctness correlation peaks near $0.19/0.87$ during the transition. These summaries in Figure 4 describe how instances differ over training; computing each instance's mode from its answer probabilities identifies the voting limit directly.

E.2 Full calibration-budget comparison

Table 6: All calibration-selected budgets at addition step 1800 and multiplication step 2000. Cells report three-seed mean accuracy / mean draws. Bold marks the highest accuracy and fewest draws in each row. Infeasible means that at least one seed has no candidate within the calibration budget. The budget limits calibration cost, not realized evaluation cost.

Budget B	CASC	ASC/Beta	ESC
	Accuracy (%) / mean draws		
Addition			
4	infeasible	**52.15** / 1.42	**52.15** / **1.41**
8	**58.12** / 4.68	52.15 / **1.42**	57.70 / 3.51
16	60.06 / 7.40	**61.66** / 9.91	60.22 / **6.16**
24	61.79 / 16.02	**62.01** / 13.60	60.22 / **6.16**
40	**62.39** / 22.85	62.23 / **18.64**	62.14 / 24.80
Multiplication			
4	**73.46** / 3.30	73.11 / 2.94	72.31 / **2.64**
8	**75.06** / 4.30	74.00 / 4.95	72.31 / **2.64**
16	78.92 / **6.52**	**80.32** / 9.88	78.95 / 11.75
24	80.41 / **9.86**	**80.99** / 12.45	78.98 / 14.66
40	80.41 / **9.86**	**80.99** / 12.45	80.93 / 21.70

CASC improves on both reported means relative to ASC/Beta at multiplication $B = 8$, and relative to ESC at multiplication $B = 24$ and addition $B = 40$. ESC improves on both means relative to CASC at addition $B = 16$, and ASC/Beta does so at addition $B = 24$. At multiplication $B = 16$, CASC uses 44.6% fewer draws than ESC with accuracy lower by 0.03 percentage points; the accuracy difference is small, although we did not test equivalence. The remaining feasible comparisons trade accuracy against draw cost.

E.3 Ablating the systematic-error component

To isolate the effect of systematic-error discounting, the ablation sets only α to zero across all 84 checkpoints. It retains q, r, ε, answer streams, vote ties, odd-step checks and $K = 40$, with abstention disabled. Fitted $q \le r$ occurs at 21 of the 84 checkpoints, where, once a second answer appears, the leader's score no longer favors larger counts. The comparison spans six training runs; checkpoints within a run are repeated observations. Numerical checks recover all 756 full-policy predictions and stopping times across settings and checkpoints; an independent evaluation of Eq. 7 agrees within 8×10^{-15}.

Table 7: Systematic-component ablation at fixed $\delta = .05$. Only α changes; entries are evaluation mean $\pm$ sample standard deviation over three training seeds. Retaining the term adds draws for small observed accuracy gains.

	Addition @ 1800		Multiplication @ 2000	
Variant	Acc. (%) ↑	Draws ↓	Acc. (%) ↑	Draws ↓
CASC	62.39±36.62	21.66±18.49	80.29±24.35	11.76±12.53
CASC, $\alpha = 0$	62.23±36.59	19.71±16.49	80.19±24.51	10.89±11.02

At the two transition checkpoints (Table 7), retaining α changes stopping times on $9.22\%/5.58\%$ of evaluation instances, but changes final answers on only $0.61\%/0.32\%$. Full CASC minus the ablated variant has mean accuracy differences of $+0.16/+0.10$ percentage points, with paired-seed 95% Student-t intervals $[-0.74, 1.06]/[-0.32, 0.51]$. Its mean draw differences are $+1.94/+0.87$. At these checkpoints the systematic term therefore delays some unanimous prefixes considerably while rarely changing their final answers, and three seeds show no clear accuracy gain.

For the budget-conditioned comparison, each variant selects its own δ from the same nine-setting grid, using calibration labels only and the same accuracy, cost and tie-breaking criteria as the main comparison. Table 8 reports all five budgets. Removing α makes addition at $B = 4$ feasible and changes the selected settings at some larger budgets. Because retuning can choose a stricter threshold, the ablated policy may use more evaluation draws at a common budget, as on multiplication at $B = 8$ and 16. This is consistent with the fixed-threshold ordering in Appendix B.2. Neither variant uniformly improves accuracy and cost across these calibration-selected points.

Table 8: Calibration-selected systematic-component ablation at the same transition checkpoints. Cells are three-seed mean evaluation accuracy / mean draws. Each variant independently selects its threshold on calibration; infeasible means no eligible setting in at least one seed.

Budget B	CASC	CASC, $\alpha = 0$
	Accuracy (%) / mean draws	
Addition		
4	infeasible	54.77 / 3.11
8	58.12 / 4.68	58.15 / 3.99
16	60.06 / 7.40	60.10 / 6.71
24	61.79 / 16.02	61.50 / 15.06
40	62.39 / 22.85	62.23 / 20.76
Multiplication		
4	73.46 / 3.30	73.46 / 3.30
8	75.06 / 4.30	78.92 / 5.94
16	78.92 / 6.52	80.10 / 8.11
24	80.41 / 9.86	80.35 / 9.22
40	80.41 / 9.86	80.35 / 9.22

E.4 A CORRECTNESS-CALIBRATED CONFIDENCE CONTROL

Raw ECE compares scores with different purposes and access to labels: ASC/Beta estimates vote stability, whereas CASC uses labeled calibration to construct a correctness score. We therefore apply the same post-hoc correctness map to both policies. The policies remain at $c = .95$ and $\delta = .05$, with full coverage; their answers and stopping times remain unchanged. For each seed and checkpoint, the map uses the same 1,044 calibration instances and evaluates it on the disjoint 1,045 instances. No evaluation labels select the mapping or its parameters.

Partition the original score range into ten fixed bins $[0, .1), \ldots, [.9, 1]$. For a bin with $m_b > 0$ calibration instances and s_b correct predictions, assign the smoothed frequency $g_b = (s_b+1)/(m_b+2)$. An empty bin receives the global smoothed calibration accuracy. We keep this control simple so that it measures only what correctness labels add to each score, leaving both stopping rules unchanged. We analyze all 84 checkpoints and retain the same two transition checkpoints for the detailed comparison in Table 9.

The mapping removes most of the raw ECE gap. Mapped CASC minus mapped ASC/Beta Brier differences are 0.0019 on addition and 0.0116 on multiplication, with paired-seed 95% Student-t intervals $[-0.0085, 0.0122]$ and $[-0.0196, 0.0428]$. Both intervals include zero. Because the two policies can predict different answers, each map scores confidence in that policy's own decisions.

A low ECE can coexist with poor discrimination between instances. As a reference, assigning every instance its policy's global smoothed calibration accuracy already yields ECE $0.013/0.019$ for ASC/Beta and $0.010/0.029$ for CASC. This constant-score reference has Brier scores $0.145/0.109$ and $0.145/0.120$, respectively, versus the mapped scores in Table 9. Both maps provide only small observed Brier improvements over that reference. Correctness calibration and adaptive sampling

Table 9: Correctness calibration with unchanged policy decisions. Brier score is mean squared error between confidence and correctness, reported as three-seed mean ± sample standard deviation. ECE pools the three seeds into ten fixed-width confidence bins. Lower is better for both metrics. Both correctness maps use calibration labels only. Bold marks the lowest error in each column.

	Addition @ 1800		Multiplication @ 2000	
Score	Brier ↓	ECE ↓	Brier ↓	ECE ↓
ASC/Beta (raw)	0.280±0.271	0.275	0.144±0.191	0.115
CASC (raw)	0.193±0.167	0.187	0.171±0.201	0.166
ASC/Beta + correctness map	**0.139**±0.120	**0.011**	**0.103**±0.120	**0.026**
CASC + correctness map	0.141±0.120	0.014	0.115±0.108	0.030

should therefore be evaluated separately. We do not test whether remapped scores would improve an online stopping rule or transfer to another checkpoint.

E.5 Selective prediction on arithmetic streams

Instances the gate can reject. For $K > 3$ and $\gamma > 0$, every CASC rejection is contained in $U_3 = \{Y_1 = Y_2 = Y_3\}$, because once a second answer appears, $\pi_n = 0$ for the rest of the stream. With $0 \le \alpha < 1$ and $0 < q < 1$, write $u_3 = \alpha/[\alpha + (1-\alpha)q^3]$ for the unanimous-prefix score. If $u_3 \le \delta$ and $u_3 < \gamma$, every unanimous prefix commits at its first check, and no instance can subsequently abstain. The second condition is strict because rejection is checked before commitment. More generally, a surviving unanimous prefix has increasing systematic score until a second answer appears or it stops. This explains why a low fitted α can disable rejection despite nonzero error on arithmetic streams.

Calibration-selected gates and common-coverage controls. We evaluate selective prediction at the six transition checkpoints with $\delta = .05$ and $\gamma \in \{.05, .1, .2, .3, .4, .5, .6, .7, .8, .9, .95, .99, 1.01\}$. For each seed, calibration maximizes committed accuracy subject to minimum coverage $C_{\min} \in \{.7, .9\}$; ties prefer higher coverage, then fewer draws. The evaluation split uses the selected gate unchanged. To compare rejection at the same coverage, we rank the full-coverage CASC and ASC/Beta outputs by the correctness maps in Appendix E.4. For each seed, retain the highest mapped scores at the gate's realized evaluation coverage, sharing a boundary tie's retention probability equally. These controls use evaluation scores and the achieved coverage, never evaluation correctness; fractional retention integrates over random ties. They are retrospective ranking comparisons and carry no coverage guarantee for deployment. Random rejection of full-coverage CASC is an additional reference whose expected accuracy equals its full-coverage accuracy.

Table 10: Real-stream selective prediction. Entries are three-seed mean ± sample standard deviation. All rows share each seed's expected retained fraction: 99.01% on addition and 100% on multiplication on average. Both minimum-coverage constraints select identical gates. "Score" uses the calibration-only correctness map; draws count every instance, including rejections. The gate retains nearly the full population.

	Addition @ 1800		Multiplication @ 2000	
Retention rule	Acc. (%) ↑	Draws ↓	Acc. (%) ↑	Draws ↓
CASC γ gate	62.43±36.57	21.31±17.98	80.29±24.35	11.76±12.53
CASC + random	62.39±36.62	21.66±18.49	80.29±24.35	11.76±12.53
CASC + score	62.39±36.62	21.66±18.49	80.29±24.35	11.76±12.53
ASC/Beta + score	62.34±36.66	18.49±12.13	81.15±25.82	14.14±9.15

Both coverage constraints select $\gamma = (.95, .05, 1.01)$ for addition seeds $(0, 1, 2)$ and $(.05, .05, 1.01)$ for multiplication. The three seeds with $\gamma = .05$ satisfy the no-rejection condition above; the two with 1.01 explicitly disable abstention. Only addition seed 0 rejects, retaining 97.03% of evaluation instances. Every retained prediction and stopping time equals its full-coverage CASC counterpart. The gate's accuracy minus mapped ASC/Beta at the same coverage is $+0.09/-0.86$ percentage

points, with paired-seed 95% Student-t intervals $[-0.77, 0.95]/[-4.63, 2.91]$. On these arithmetic streams, Table 10 therefore shows neither the large synthetic abstention gain nor an improvement over a selector that uses the same labels.

E.6 Selective prediction on synthetic streams

CASC's abstention threshold γ tests whether the systematic-error score can identify instances to reject. Appendix Figure 7(b) tests the mechanism on streams generated by the SSM ($\alpha = .30, q = .72$): as coverage drops from 1.0 to 0.69, accuracy on answered instances rises from 0.689 (near the 0.70 sampling ceiling) to 0.989, while the Beta rule at threshold .95 must answer everything at 0.697. The full-coverage reference only illustrates the effect of rejection and is not a matched-coverage comparison. The simulation shows that this score can reject much of the systematic (always wrong) component, but it does not validate posterior calibration or show that selective accuracy increases monotonically with the threshold. The arithmetic comparison in Appendix E.5 finds much less rejection.

E.7 Predictions under the mixture model

Correlation and premature stopping. At $\alpha = .25, q = .70, n = 21$, Eq. 3 predicts sample-mean variance 0.0994, versus empirical 0.0993. The i.i.d. prediction is 0.0119, smaller by a factor of 8.4, and $n_{\text{eff}}(21) = 2.51$. With $\alpha = .30, q = .72$, Beta stopping uses 7.5 draws on average and attains accuracy 0.6997, close to the binary-majority fixed-40 benchmark 0.6988. On the 30% systematic component, it stops after exactly four draws at nominal confidence 0.969 and is always wrong. Deterministic checks of the marginal gains and moment inversion agree to 10^{-10}; oracle greedy and DP agree at all 16 tested budgets.

The synthetic results connect correlation, premature commitment and sampling gains. The curves below show how accuracy and cost vary across stopping settings; the budget comparison in Table 6 selects settings independently on calibration data.

Accuracy and sampling cost. Figure 6 shows the operating points available as stopping thresholds or window sizes change. The curves describe the shared evaluation streams; the independent calibration rule in Table 6 selects the reported budget-conditioned settings. A favorable segment therefore reflects a tradeoff between accuracy and cost at particular settings, and its ordering may change at other budgets. Greedy remains a reference with access to the full distribution.

Calibration and selective answers. Figure 7 separates two questions. The arithmetic reliability plot compares confidence with observed correctness; its raw gap depends on the checkpoint and largely closes after remapping with the same calibration labels. The SSM simulation instead changes which examples receive an answer. Its gain in committed accuracy comes with reduced coverage, so on its own it says nothing about how it compares with another selective policy. The corresponding arithmetic control is in Table 10.

F Stopping-Policy Comparisons on Public GSM8K Responses

Data and model identity. The comparison isolates stopping decisions by giving every policy the same publicly available responses. All answers come from the official ESC release; this offline evaluation makes no model API calls. The data are its three `T0.7.jsonl` GSM8K response banks, commit `27574d4`.[4] The upstream paper identifies the models as GPT-3.5-Turbo, GPT-4 and Llama-2 7B, with few-shot prompts and temperature 0.7; it reports top-$p = 0.9$ for Llama-2 (Li et al., 2024). Model names follow the source release, which does not identify exact model snapshots or generation dates for individual responses. The results therefore characterize stopping on these historical answer banks, with no new generation from model endpoints.

All gold explanations match the official GSM8K test file at commit `3101c7d`. We align questions after removing only an initial `Q:` wrapper and surrounding whitespace. GPT-3.5 lacks one of the 1,319 official questions; all three models use the same remaining 1,318, with no duplicate questions

[4] Fixed-version answer files: GPT-3.5, GPT-4, and Llama-2 7B.

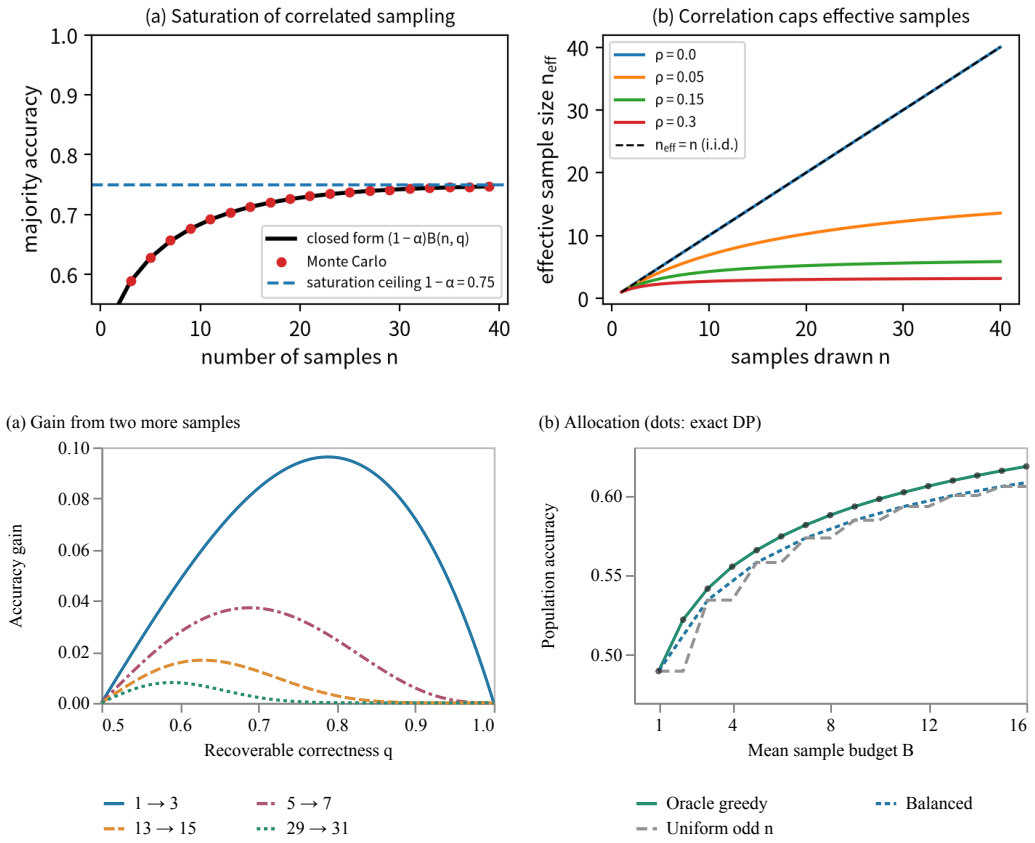


Figure 5: **Theory validation (synthetic).** Top: closed-form SSM curves match Monte-Carlo and saturate at $1 - \alpha$; correlation caps effective sample size. Bottom: gains from an extra sample pair peak strictly above $q = .5$, with the peak approaching the boundary as n grows. Oracle greedy allocation on 16 equally weighted types matches the exact DP (dots). Balanced allocation uses B draws per query for odd B; for even B, eight uniformly selected queries receive $B - 1$ and the others $B + 1$, independently of types and answers. It spends exactly $16B$ draws; accuracy averages analytically over the assignment. Uniform odd-n sampling can leave budget unused.

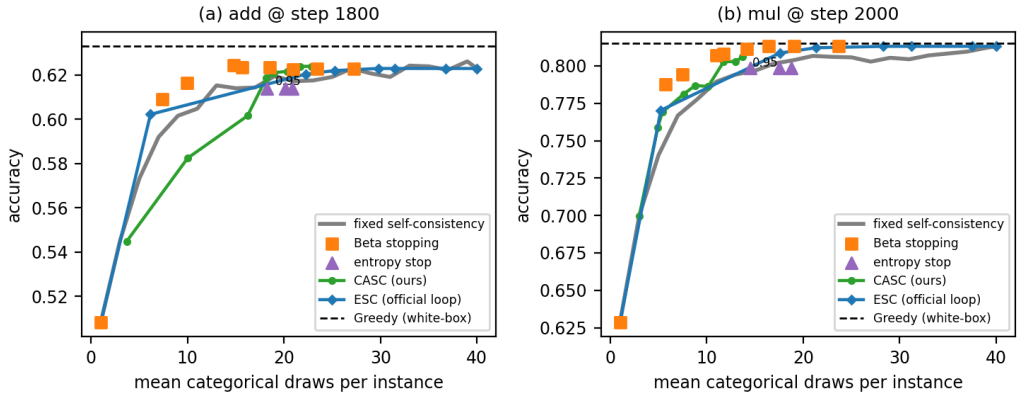


Figure 6: Accuracy and mean sample count across stopping settings, averaged over the same three training seeds. ESC uses its official rule with eight window sizes. The dashed Greedy line is a white-box reference, not an additional sampling policy. No point on these evaluation curves is used to tune the reported default settings.

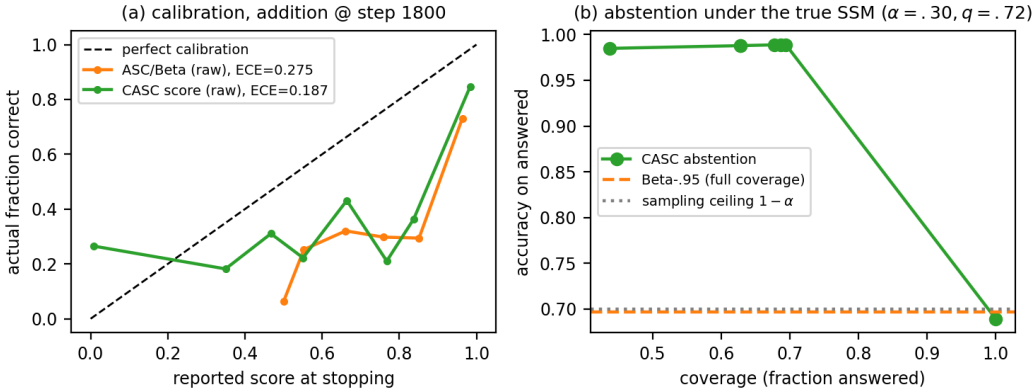


Figure 7: (a) Raw correctness calibration at addition step 1800: CASC has lower ECE at this checkpoint; the ordering reverses at multiplication step 2000. Calibration-only remapping largely closes this gap (Table 9). (b) The synthetic abstention experiment on SSM-generated streams trades coverage for committed accuracy. The full-coverage Beta reference cannot be compared directly with methods that are allowed to reject examples.

and 40 responses each. NumPy's PCG64 generator with seed 7 partitions the common questions, in official order, into 659 calibration and 659 evaluation examples. For each question, five permutations use seeds (s, i), where $s \in \{0, 1, 2, 3, 4\}$ and i is its official index. Each method receives the same order indices. Permutations reuse the same bank and are not independent generation trials.

Answer representation and baseline decisions. Every method uses the released ESC parser: take the last numeric match and strip its trailing period; a failed parse is a distinct invalid category. The parser ignores signs in generated numbers and treats comma and decimal formatting literally. Reported accuracy uses this shared extraction rule, not semantic answer equivalence. Failed parses remain in the denominator: 30, 3 and 332 among 52,720 responses for GPT-3.5, GPT-4 and Llama-2, respectively. Gold numbers come from the dataset's marked answer field. Per-question integer encoding preserves answer equality and first-appearance ties; stopping rules receive no labels.

ASC/Beta executes `BetaStoppingCriteria.should_stop` from the official AdaptiveConsistency repository at commit `899ab56`. The rule depends on the top-two counts and threshold; caching these quantities preserves its probability and stopping decision. The leader is determined

by the actual prefix. All 3,960 reachable count/threshold checks through 40 votes agree in stopping decision with the incomplete-beta implementation; all predictions and stopping times also agree on the public answer streams. ESC uses the official rule that returns a unanimous window's answer or the plurality at the cap. The text and integer representations yield the same decisions. Greedy decoding requires probabilities unavailable in these answer banks, so this comparison excludes it.

Calibration and the systematic-error component. We fit CASC once per model using each calibration question's 40 responses. The fitted $(\hat{\alpha}_K, \hat{q}, \hat{r})$ are $(.0319, .7939, .1021)$, $(.0334, .9028, .1744)$ and $(.5053, .3964, .1859)$ for GPT-3.5, GPT-4 and Llama-2. Thus Llama-2 is outside SSM's $q > 1/2$ regime, and GPT-4 has $q + r > 1$, so the novelty weight takes its minimum value. The fitted values are therefore heuristic scoring weights and do not define a valid SSM probability model. The policy uses $M = 41$ and full coverage ($\gamma = 1.01$). Changing only M to 59 leaves all predictions and stopping times unchanged on these banks; maximum score discrepancy is 7.1×10^{-15}. The component control sets only $\alpha = 0$, retaining fitted q, r and the same stopping grid. All default accuracy differences between CASC and this control have intervals containing zero. On Llama-2 the control uses 21.24 rather than 25.24 answers at the same rounded accuracy.

Budget selection and uncertainty. SC uses odd counts from 1 to 39 and the cap 40; ASC/Beta uses thresholds $\{.75, .8, .85, .9, .925, .95, .975, .99, .999\}$; ESC uses windows $\{1, 2, 4, 5, 8, 10, 20, 40\}$; CASC and its control use $\delta \in \{.7, .6, .5, .3, .2, .1, .05, .02, .01\}$. For $B \in \{4, 8, 16, 24, 40\}$, each family selects the highest calibration accuracy among settings with mean calibration draws at most B. Ties favor fewer draws, then lexicographic configuration key. Calibration metrics average all five orders before selecting a setting; evaluation accuracy never enters selection. The shared calibration banks contain $659 \times 40 = 26,360$ responses per model, excluded from online counts. Tuned baselines receive the same labels; CASC additionally uses them for its population fit. Actual evaluation cost can exceed B.

Intervals use 2,000 paired bootstrap resamples of the 659 evaluation questions (PCG64 seed 1729), after averaging each question over orders. They condition on this calibration split and these fixed response banks; they are pointwise and exploratory, and they do not cover simultaneous inference or the variability of fresh generations. Tables 11 to 15 retain all model/budget combinations and method comparisons, including CASC's infeasible $B = 4$ setting on Llama-2.

Availability, stopping and cost. At least one response is correct on 98.33%, 97.42% and 46.13% of evaluation questions for GPT-3.5, GPT-4 and Llama-2; SC@40 accuracy is 85.92%, 89.56% and 19.51%. These finite-bank figures are diagnostics and should not be read as exact modal or infinite-sampling ceilings. At $B = 8$, GPT-3.5 CASC improves on fixed SC in both accuracy and draws, while Beta reaches higher accuracy with more draws. At $B = 16$, Llama-2 ESC is both more accurate and cheaper than CASC. The preferred policy thus depends on the model and budget. Answer counts quantify sampling efficiency; token cost and end-to-end latency require separate measurements.

Table 11: Calibration-budget selections from ESC's public GSM8K answer banks. Each cell gives accuracy (%) / actual mean draws. † marks an evaluation mean above calibration budget B; a dash denotes the absence of an eligible CASC setting. Settings are selected separately within each model/family using only calibration data. Bold marks the highest accuracy and fewest draws within each model and budget, including ties.

B	SC	ASC/Beta	ESC	CASC	CASC, $\alpha = 0$
GPT-3.5					
4	79.18 / **3.00**	**84.40** / 4.14†	83.61 / 3.82	82.64 / 3.73	82.64 / 3.73
8	83.25 / 7.00	**85.80** / 7.67	83.61 / **3.82**	85.10 / 5.01	85.64 / 6.12
16	84.58 / 13.00	**85.86** / 12.19	85.74 / 15.37	85.10 / **5.01**	85.64 / 6.12
24	85.34 / 23.00	**85.92** / 19.17	**85.92** / 21.46	85.71 / 21.45	85.64 / **6.12**
40	85.71 / 25.00	**85.92** / 19.17	**85.92** / 21.46	85.71 / 21.45	85.64 / **6.12**
GPT-4					
4	89.14 / 3.00	**89.65** / 2.79	89.53 / **2.33**	89.41 / 3.04	89.41 / 3.04
8	**89.92** / 7.00	89.53 / 7.61	89.44 / 8.24†	89.41 / **3.04**	89.41 / **3.04**
16	**89.92** / 7.00	89.53 / 7.61	89.56 / 12.98	89.41 / **3.04**	89.41 / **3.04**
24	**89.86** / 23.00	89.53 / 7.61	89.56 / 12.98	89.41 / **3.04**	89.41 / **3.04**
40	**89.56** / 40.00	89.53 / 7.61	**89.56** / 12.98	89.41 / **3.04**	89.41 / **3.04**
Llama-2 7B					
4	**17.81** / 3.00	17.12 / **1.00**	17.12 / **1.00**	—	**17.81** / 3.00
8	**18.57** / 7.00	17.12 / **1.00**	17.12 / **1.00**	17.81 / 6.05	**18.57** / 6.76
16	19.00 / 13.00	19.36 / 13.59	19.30 / **8.48**	18.57 / 9.81	**19.54** / 15.46
24	19.27 / 21.00	19.45 / 24.11†	19.30 / **8.48**	**19.58** / 21.64	**19.58** / 18.24
40	19.51 / 40.00	19.45 / 24.11	19.54 / 31.28	19.48 / 27.97	**19.58 / 18.24**

Table 12: Default CASC minus each comparator on ESC's public GSM8K answer banks. Accuracy differences are percentage points (pp); positive draw differences mean CASC costs more. Brackets give paired question-bootstrap 95% intervals. These are operating-point comparisons, not equal-cost comparisons.

Comparator	Δ acc. [95% CI]	Δ draws [95% CI]
GPT-3.5		
SC@40	−0.79 [−1.49, −0.09]	−19.60 [−20.67, −18.45]
ASC/Beta ($c = .95$)	−0.73 [−1.43, −0.06]	+10.27 [+8.74, +11.90]
ESC ($w = 5$)	−0.61 [−1.31, +0.06]	+5.02 [+3.20, +6.96]
CASC, $\alpha = 0$	+0.03 [−0.12, +0.21]	+15.28 [+14.10, +16.55]
GPT-4		
SC@40	+0.27 [−0.33, +0.94]	−36.33 [−36.46, −36.19]
ASC/Beta ($c = .95$)	+0.18 [−0.33, +0.73]	−2.49 [−2.86, −2.14]
ESC ($w = 5$)	+0.39 [−0.18, +1.00]	−4.57 [−5.06, −4.09]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]
Llama-2 7B		
SC@40	−0.03 [−0.18, +0.09]	−14.76 [−15.64, −13.87]
ASC/Beta ($c = .95$)	+0.03 [−0.09, +0.15]	+1.13 [+0.39, +1.82]
ESC ($w = 5$)	−0.06 [−0.21, +0.09]	−6.04 [−7.00, −5.07]
CASC, $\alpha = 0$	+0.00 [−0.15, +0.15]	+4.00 [+3.37, +4.62]

Table 13: GPT-3.5: calibration-selected CASC minus each selected comparator. Accuracy differences are percentage points; brackets are paired question-bootstrap 95% intervals. Actual costs appear in Table 11.

Comparator	Δ acc. [95% CI]	Δ draws [95% CI]
Budget $B = 4$		
SC	+3.46 [+2.58, +4.40]	+0.73 [+0.54, +0.93]
ASC/Beta	−1.76 [−2.49, −1.06]	−0.41 [−0.63, −0.20]
ESC	−0.97 [−1.73, −0.30]	−0.09 [−0.28, +0.07]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]
Budget $B = 8$		
SC	+1.85 [+1.12, +2.67]	−1.99 [−2.28, −1.71]
ASC/Beta	−0.70 [−1.40, −0.03]	−2.67 [−3.06, −2.24]
ESC	+1.49 [+0.82, +2.15]	+1.18 [+1.03, +1.33]
CASC, $\alpha = 0$	−0.55 [−1.09, −0.06]	−1.12 [−1.32, −0.93]
Budget $B = 16$		
SC	+0.52 [−0.18, +1.18]	−7.99 [−8.28, −7.71]
ASC/Beta	−0.76 [−1.52, −0.03]	−7.18 [−7.80, −6.53]
ESC	−0.64 [−1.43, +0.12]	−10.37 [−11.19, −9.54]
CASC, $\alpha = 0$	−0.55 [−1.09, −0.06]	−1.12 [−1.32, −0.93]
Budget $B = 24$		
SC	+0.36 [−0.18, +0.88]	−1.55 [−2.58, −0.45]
ASC/Beta	−0.21 [−0.67, +0.24]	+2.28 [+0.58, +4.08]
ESC	−0.21 [−0.67, +0.24]	−0.01 [−1.87, +1.98]
CASC, $\alpha = 0$	+0.06 [−0.09, +0.21]	+15.33 [+14.15, +16.60]
Budget $B = 40$		
SC	+0.00 [−0.61, +0.58]	−3.55 [−4.58, −2.45]
ASC/Beta	−0.21 [−0.67, +0.24]	+2.28 [+0.58, +4.08]
ESC	−0.21 [−0.67, +0.24]	−0.01 [−1.87, +1.98]
CASC, $\alpha = 0$	+0.06 [−0.09, +0.21]	+15.33 [+14.15, +16.60]

Table 14: GPT-4: calibration-selected CASC minus each selected comparator, with the same metrics and interval construction as Table 13. The low-cost CASC selection is identical at all five budgets, while competing settings can change.

Comparator	Δ acc. [95% CI]	Δ draws [95% CI]
Budget $B = 4$		
SC	+0.27 [+0.06, +0.55]	+0.04 [+0.02, +0.05]
ASC/Beta	−0.24 [−0.82, +0.33]	+0.24 [+0.05, +0.43]
ESC	−0.12 [−0.61, +0.36]	+0.70 [+0.64, +0.76]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]
Budget $B = 8$		
SC	−0.52 [−1.06, −0.03]	−3.96 [−3.98, −3.95]
ASC/Beta	−0.12 [−1.12, +0.82]	−4.58 [−5.14, −4.05]
ESC	−0.03 [−0.97, +0.91]	−5.21 [−5.80, −4.62]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]
Budget $B = 16$		
SC	−0.52 [−1.06, −0.03]	−3.96 [−3.98, −3.95]
ASC/Beta	−0.12 [−1.12, +0.82]	−4.58 [−5.14, −4.05]
ESC	−0.15 [−1.18, +0.82]	−9.94 [−10.72, −9.18]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]
Budget $B = 24$		
SC	−0.46 [−1.21, +0.30]	−19.96 [−19.98, −19.95]
ASC/Beta	−0.12 [−1.12, +0.82]	−4.58 [−5.14, −4.05]
ESC	−0.15 [−1.18, +0.82]	−9.94 [−10.72, −9.18]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]
Budget $B = 40$		
SC	−0.15 [−1.18, +0.82]	−36.96 [−36.98, −36.95]
ASC/Beta	−0.12 [−1.12, +0.82]	−4.58 [−5.14, −4.05]
ESC	−0.15 [−1.18, +0.82]	−9.94 [−10.72, −9.18]
CASC, $\alpha = 0$	+0.00 [+0.00, +0.00]	+0.00 [+0.00, +0.00]

Table 15: Llama-2: calibration-selected CASC minus each selected comparator, with the same metrics and intervals. At $B = 4$, CASC has no eligible setting, so no paired difference is defined.

Comparator	Δ acc. [95% CI]	Δ draws [95% CI]
Budget $B = 4$		
SC	—	—
ASC/Beta	—	—
ESC	—	—
CASC, $\alpha = 0$	—	—
Budget $B = 8$		
SC	-0.76 [-1.49, -0.06]	-0.95 [-1.58, -0.31]
ASC/Beta	$+0.70$ [$+0.00$, $+1.46$]	$+5.05$ [$+4.42$, $+5.69$]
ESC	$+0.70$ [$+0.00$, $+1.46$]	$+5.05$ [$+4.42$, $+5.69$]
CASC, $\alpha = 0$	-0.76 [-1.64, $+0.03$]	-0.71 [-1.58, $+0.12$]
Budget $B = 16$		
SC	-0.42 [-1.18, $+0.33$]	-3.19 [-3.87, -2.49]
ASC/Beta	-0.79 [-1.61, $+0.03$]	-3.78 [-4.80, -2.83]
ESC	-0.73 [-1.37, -0.09]	$+1.33$ [$+0.57$, $+2.08$]
CASC, $\alpha = 0$	-0.97 [-1.91, -0.06]	-5.65 [-6.70, -4.66]
Budget $B = 24$		
SC	$+0.30$ [-0.30, $+0.94$]	$+0.64$ [-0.26, $+1.55$]
ASC/Beta	$+0.12$ [-0.18, $+0.46$]	-2.47 [-3.35, -1.65]
ESC	$+0.27$ [-0.61, $+1.12$]	$+13.16$ [$+12.44$, $+13.88$]
CASC, $\alpha = 0$	$+0.00$ [-0.12, $+0.12$]	$+3.40$ [$+2.76$, $+4.05$]
Budget $B = 40$		
SC	-0.03 [-0.15, $+0.06$]	-12.03 [-12.88, -11.18]
ASC/Beta	$+0.03$ [-0.06, $+0.12$]	$+3.85$ [$+3.22$, $+4.48$]
ESC	-0.06 [-0.21, $+0.09$]	-3.32 [-4.19, -2.43]
CASC, $\alpha = 0$	-0.09 [-0.46, $+0.24$]	$+9.73$ [$+9.12$, $+10.34$]

G Further Connections to Recent Test-Time Scaling Research

Coverage, selection and heterogeneous difficulty. Sampling coverage and voting accuracy can scale differently. Brown et al. (2024) find that answer selection can plateau while coverage grows; Schaeffer et al. (2025) explain coverage power laws through heterogeneous success probabilities. Bay & Yearick (2026) relate modal and correlation ceilings to allocation for coverage, selection and estimation. SSM makes this distinction tractable with two error regimes, and our moment diagnostic tests whether learned populations satisfy this restriction.

Objectives of allocation. Difficulty-aware allocation (Wang et al., 2025) and PETS (Liu et al., 2026) allocate samples unevenly across questions. PETS targets agreement with the infinite-budget vote; the SSM oracle targets expected correctness with known query parameters. BEST-Route (Ding et al., 2025) jointly selects the model and sample count, while CASC holds the model fixed. The different objectives and information available determine which allocation guarantees apply.

Verification and candidate selection. Zhao et al. (2025) compare responses through self-verification. Setlur et al. (2025) establish conditions under which verification-based learning improves on verifier-free trace distillation; Huang et al. (2025a) analyze coverage requirements and reward hacking in best-of-N selection with imperfect reward models. These methods change how candidates are judged. CASC instead uses answer frequencies and population calibration; our comparisons isolate stopping decisions on shared answer streams.

Confidence for ranking, stopping and abstention. Calibration relates stated confidence to empirical correctness (Guo et al., 2017; Wang et al., 2024; Huang et al., 2025b). CISC (Taubenfeld et al., 2025) uses confidence to rank answers within a question, and ReASC (Kim et al., 2026a) uses it to decide when to stop. Reasoning-oriented training can improve verbalized calibration while weakening recognition of knowledge boundaries (Zeng et al., 2025). Jurayj et al. (2025) study accuracy as answer coverage changes. We distinguish these uses by calibrating scores with decisions unchanged and comparing rejection policies at the same coverage.

Validity under repeated inspection. Confidence sequences give uncertainty guarantees simultaneously over time (Howard et al., 2021). Certified Self-Consistency (Cordero-Encinar & Duncan, 2025) and CITE (Ota et al., 2026) give sequentially valid tests for identifying the mode of the response distribution. CASC uses approximate scores without these guarantees. Even a valid certificate of the mode says nothing about correctness unless the mode is linked to the true answer.

Reasoning depth and response count. Budget forcing in s1 (Muennighoff et al., 2025) controls reasoning length; TokenSkip (Xia et al., 2025) removes lower-importance tokens, and LightThinker (Zhang et al., 2025) compresses intermediate thoughts. Self-assessed confidence (Kim et al., 2026b), sequence-level entropy (Sharma & Chopra, 2025) and reflective redundancy (Sun et al., 2026) guide when to end a trace. CASC decides whether to request another completed response. Reducing the response count therefore need not proportionally reduce token consumption or latency.

Training and the sampling distribution. DeepSeek-R1 (DeepSeek-AI, 2025) develops reasoning through reinforcement learning. Tang et al. (2025) optimize multi-sample objectives including pass@k and majority voting; Qu et al. (2025) optimize test-time compute through meta-reinforcement learning. Consistency can also guide self-training (Lee et al., 2025). Yue et al. (2025) distinguish improved sampling at small budgets from changes in coverage at large budgets. Our analysis holds the sampling law fixed at each training stage; learning can change its voting ceiling.

Diversity, search and refinement. Prompt diversity interacts with best-of-N selection and majority voting (Wang et al., 2026). Forest-of-Thought (Bi et al., 2025) combines reasoning trees with self-correction, and Bilal et al. (2026) refine trajectories before aggregation. By changing candidates or their selection, these interventions can alter the limiting vote. They complement our analysis of a given sampling law.