WHEN SHOULD SELF-CONSISTENCY TRUST TRAJECTORY SIGNALS? COMPETITION-AWARE RESELECTION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-consistency (SC) is a strong and widely used aggregation method for LLM reasoning, improving accuracy by majority voting over multiple sampled trajectories. We find that SC selection becomes substantially less reliable under strong competition between the two leading answer clusters, where additional evidence is needed to distinguish the correct candidate. By examining probability-based trajectory signals, we further find that correct answer clusters tend to have higher mean trajectory signals than competing incorrect clusters. Based on these findings, we propose **CARS** (***C**ompetition-**A**ware **R**eselection for **S**elf-Consistency*), which uses trajectory-level signals to revise SC's selection between the Top-2 candidates only under strong competition. Across the evaluated models and mathematical and scientific reasoning benchmarks, CARS achieves improvements over Vanilla SC and Trajectory-Signal-Weighted Self-Consistency on the vast majority of tasks. Further cross-model, cross-benchmark, and sensitivity analyses show that the competition-dependent gain pattern remains robust across settings and is not tied to a particular trajectory signal or gating threshold.

1 INTRODUCTION

Scaling test-time computation has become an effective way to improve the reasoning performance of large language models (LLMs) (Snell et al., 2025). By sampling multiple reasoning trajectories, it exposes multiple candidate solutions, making final answer selection a central problem. Self-Consistency (SC) addresses this by grouping trajectories by final answer and selecting the most frequent candidate, effectively using candidate frequency as evidence of correctness (Wang et al., 2023). However, prior work has shown that SC reliability depends on how support is distributed among competing candidates, with frequency becoming less discriminative when leading candidates receive similar support (Feng et al., 2026; Cordero-Encinar & Duncan, 2025).

The generation process provides another source of information. Probability-based trajectory signals, such as confidence and certainty, can reflect reasoning correctness (Kang et al., 2025; Fu et al., 2026; Liu et al., 2026) and have been used in Best-of-N selection, reranking, and weighted aggregation (Kang et al., 2025; Zhou et al., 2025; Taubenfeld et al., 2025). Yet these signals are generally less reliable than frequency when used alone (Zenn & Geiping, 2026). Existing methods often combine the two through weighted aggregation (Zhou et al., 2025; Taubenfeld et al., 2025), but do not directly address when trajectory evidence should influence a frequency-based decision. This raises a central question: **when does candidate frequency become insufficient, and when should trajectory evidence be introduced?**

We first isolate candidate selection from candidate coverage using instances where the correct answer is among the Top-2 candidates. Even under this control, SC accuracy decreases as competition between the two leading clusters increases. Meanwhile, correct candidates tend to exhibit higher mean trajectory signals than competing incorrect candidates. However, applying trajectory-based Top-2 reselection to all instances performs worse than Vanilla SC. Analysis reveals a competition-dependent pattern: frequency is more reliable when one candidate has a clear support advantage, whereas trajectory evidence becomes increasingly useful as the Top-2 support gap narrows.

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.

Motivated by this pattern, we propose **CARS** (***C**ompetition-**A**ware **R**eselection for **S**elf-Consistency*). CARS retains Vanilla SC by default and activates trajectory-based Top-2 reselection only under sufficiently strong competition, comparing the candidates by their mean trajectory signals. Our main instantiation uses certainty over the trajectory tail, while the framework is not tied to a specific signal. Thus, **candidate frequency determines *when* auxiliary evidence is introduced, while trajectory information determines *how* to distinguish the competing candidates**.

Across multiple LLM families and mathematical and scientific reasoning benchmarks, CARS improves over Vanilla SC and Trajectory-Signal-Weighted Self-Consistency (TSWS) on the vast majority of evaluated tasks. Cross-model, cross-benchmark, and sensitivity analyses further show that the competition-dependent gain pattern is robust across trajectory signals and gating thresholds.

Our main contributions are:

1. We characterize the selection failure regime of self-consistency and show that, after controlling for candidate coverage, SC becomes less reliable under strong Top-2 competition.
2. We show that trajectory-level signals provide complementary correctness information beyond candidate frequency, with correct answer clusters tending to exhibit higher mean signals than competing incorrect clusters.
3. We propose CARS, which selectively applies trajectory-based Top-2 reselection under strong competition, yielding consistent gains across models and benchmarks and remaining robust across trajectory signals and gating thresholds.

2 Diagnosing and Addressing Self-Consistency Failures

2.1 Where Does Self-Consistency Become Unreliable?

Self-consistency (SC) is an effective aggregation rule, but its success requires two conditions: the correct answer must be sampled, and its candidate cluster must receive sufficient support to become the largest cluster. These correspond to candidate coverage and candidate selection, respectively. Prior work has shown that SC reliability is closely related to how support is distributed among competing candidates, particularly the separation between the correct answer and its strongest alternatives (Feng et al., 2026). Motivated by this, we characterize the candidate distribution through its leading clusters, considering both how much support is captured by the Top-2 candidates and how strongly they compete with each other.

Let $C_1, C_2, \ldots$ denote candidate clusters formed by final-answer string matching and ranked by support, with C_1 and C_2 being the largest and second-largest clusters, respectively, and let q_i denote the fraction of valid trajectories belonging to C_i. Vanilla SC always selects C_1. We characterize the Top-2 structure using

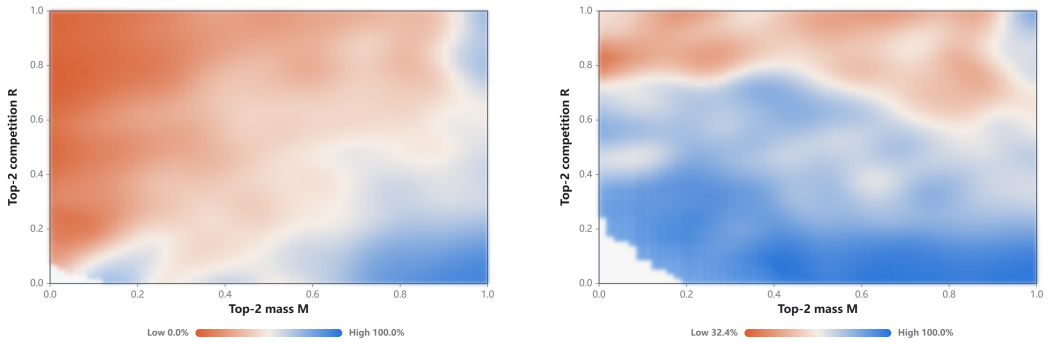
$$M = q_1 + q_2, \qquad R = \frac{q_2}{q_1}, \tag{1}$$

where M measures the support concentrated in the Top-2 candidates and R measures their competition intensity. Larger R indicates a smaller relative support gap between the two candidates.

As shown in Figure 1(a), SC performs best in the bottom-right region of the (M, R) plane, where most probability mass is concentrated in the Top-2 candidates and their competition is weak. Its accuracy decreases outside this region, as the candidate distribution becomes more dispersed or the Top-2 candidates become more competitive. Note, however, that the failures in Figure 1(a) conflate candidate coverage and candidate selection: the correct answer may either fail to reach the leading candidates or lose to a competing candidate after it has already emerged.

To isolate the reliability of SC as a selection mechanism, we construct a controlled subset containing only instances where the correct answer is already among the Top-2 candidates. Figure 1(b) shows that, under this control, SC selection accuracy depends strongly on R but much less on M. In particular, most selection failures occur under strong competition, where R approaches 1.

These results identify strong Top-2 competition as a key failure regime of SC selection. When the two leading candidates receive comparable support, frequency alone becomes insufficient for reliably distinguishing the correct candidate. This motivates us to seek additional complementary evidence to help identify the correct answer.



(a) All samples **(b)** Top-2-controlled subset

Figure 1: SC accuracy across the candidate-distribution plane defined by Top-2 mass M and Top-2 competition R. Panel (a) uses all problem instances, whereas panel (b) retains only instances for which the correct answer is among the Top-2 candidates.

2.2 Trajectory Signal as Complementary Evidence

LLM generation contains information beyond the answer frequency used by SC. Prior work has shown that probability information along reasoning trajectories can reflect trajectory correctness, and such model-side information has been used in methods such as probability-based Best-of-N selection (Kang et al., 2025). Trajectory-level probability signals may therefore provide complementary evidence when frequency alone is insufficient for candidate selection.

We consider two representative signals, *confidence* and *certainty*. For a trajectory $\tau = (x_1, \ldots, x_T)$, confidence is defined as

$$\mathrm{Conf}(\tau) = \exp\left(\frac{1}{T}\sum_{t=1}^{T}\log p_t(x_t)\right), \tag{2}$$

the geometric mean of the realized token probabilities (Zhou et al., 2025). Certainty instead reflects how concentrated the model's local next-token distribution is (Kang et al., 2025). Following prior work (Fu et al., 2026), we compute

$$c_t(\tau) = -\frac{1}{K}\sum_{v\in\mathcal{V}_t^{(K)}}\log p_t(v), \qquad \mathrm{Cert}(\tau) = \frac{1}{T}\sum_{t=1}^{T} c_t(\tau), \quad K = 20, \tag{3}$$

Here $p_t(v)$ is the sampling probability after temperature and top-p, conditioned on the prompt and $x_{<t}$; $\mathcal{V}_t^{(K)}$ contains the K most probable tokens. Confidence scores realized tokens, while certainty captures preferences over plausible continuations.

Prior work further suggests that different parts of a reasoning trajectory may contain different amounts of correctness-related information, and local signals can sometimes be more informative than whole-trajectory statistics (Fu et al., 2026). For example, signals near the end of the trajectory are more directly related to the reasoning that produces the final answer. We therefore also consider tail-based variants.

A preliminary comparison of global and local confidence and certainty signals (Appendix B.1) shows that certainty over the final 30% of the trajectory provides strong within-question discrimination. We use this as the default trajectory-level signal:

$$s_{\text{tail-30\%}}(\tau) = \frac{1}{m}\sum_{t=T-m+1}^{T} c_t(\tau), \qquad m = \lceil 0.3T \rceil, \tag{4}$$

where $c_t(\tau)$ is the token-level certainty defined above. We use $s(\tau)$ throughout the following analysis and main experiments, while Section 4.4 evaluates alternative global and local trajectory signals.

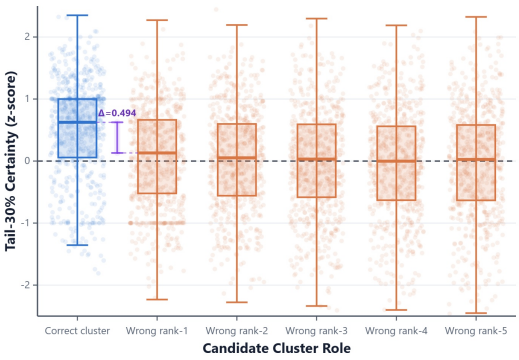


Figure 2: Within-question standardized tail-30% certainty for the largest correct and support-ranked wrong clusters.

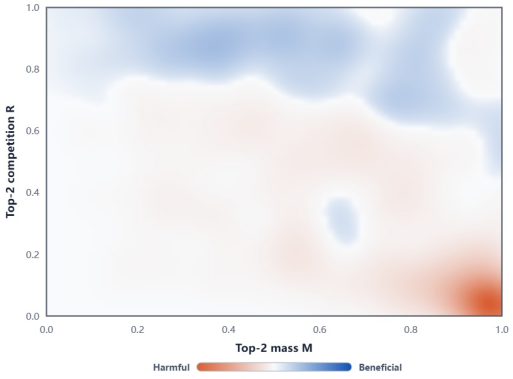


Figure 3: Smoothed signed net-count contribution of certainty-based reselection over Vanilla SC across (M, R).

2.3 Trajectory Signals for Candidate Reselection

The trajectory signal $s(\tau)$ can reflect the correctness of an individual reasoning trajectory, but SC ultimately selects among answer clusters rather than individual trajectories. This raises a natural question: when multiple trajectories support the same candidate, do their signals form a stable candidate-level difference?

To examine this, we summarize trajectories within each candidate cluster C_i by their mean signal,

$$\bar{s}_i = \frac{1}{|C_i|} \sum_{\tau \in C_i} s(\tau). \tag{5}$$

Intuitively, $\bar{s}_i$ measures how strongly the model, on average, favors the trajectories leading to candidate C_i. Unlike a summed score, it does not directly increase with cluster size, allowing trajectory-level preference to remain distinct from frequency evidence.

We then examine whether correct and incorrect candidates exhibit systematic differences in $\bar{s}_i$. Figure 2 (Appendix C.4) shows higher within-question z-scores for correct than competing incorrect clusters. Although the two distributions overlap, this consistent gap indicates that trajectory-level preference provides useful candidate-level correctness information beyond answer frequency.

This observation suggests that the mean trajectory signal can serve as an auxiliary criterion for candidate selection. In particular, when comparing the Top-2 candidates, the candidate with the larger $\bar{s}_i$ becomes a natural alternative to the frequency-based choice made by SC.

2.4 When Should Trajectory Evidence Influence the Decision?

To test the effectiveness of trajectory-based reselection, we first apply this selection rule to all instances. Despite the discriminative information carried by $\bar{s}_i$, unconditional Top-2 reselection performs worse than Vanilla SC across all three evaluated models. This shows that trajectory-based reselection is not a general replacement for frequency-based SC.

From an evidence perspective, the two selection mechanisms rely on different candidate-level information: SC uses candidate frequency q_i, whereas reselection uses trajectory evidence $\bar{s}_i$. This raises a key question: **when should we trust frequency, and when should we trust trajectory evidence?**

To study this, we again use the Top-2-controlled subset from Figure 1(b), where the correct answer is already among the two leading candidates. Over the same (M, R) plane, we examine the *rescue* and *broken* cases of reselection relative to SC. We define a rescue as an instance where SC selects the wrong candidate but reselection selects the correct one, and a broken case as one where SC is correct but reselection changes the decision to an incorrect candidate.

As shown in Figure 3, reselection produces more rescues in the high-competition region, but causes substantial broken cases in the bottom-right region where SC is already reliable. Notably, the relative

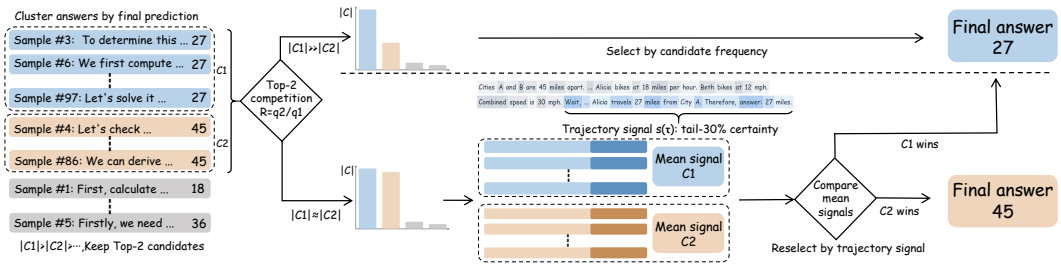


Figure 4: **Overview of CARS.** Sampled trajectories are clustered by final prediction; weak Top-2 competition retains the frequency winner, while strong competition triggers Top-2 reselection.

value of the two evidence sources depends strongly on Top-2 competition: when one candidate has a clear support advantage, frequency is more reliable; as the support gap narrows, trajectory evidence becomes increasingly useful.

The simplest implication is therefore to retain SC under weak competition and introduce trajectory-based reselection only when R is sufficiently high. This provides a simple competition-dependent principle for selectively incorporating trajectory evidence into candidate selection.

In summary, our analysis shows that SC selection becomes unreliable under strong Top-2 competition, where candidate frequency provides weaker evidence for identifying the correct answer. Trajectory-level signals provide complementary correctness information, with higher cluster-level means for correct than incorrect candidates. However, unconditional reselection does not outperform SC, indicating that the two evidence sources are reliable in different regimes. Together, these findings support a competition-dependent principle: retain frequency-based SC under weak competition and introduce trajectory evidence only under sufficiently strong Top-2 competition.

3 Competition-Aware Reselection for Self-Consistency

The analysis in Section 2 shows that the benefit of trajectory-level signals is concentrated in cases where the two leading candidate clusters receive similar support. Motivated by this observation, we propose *Competition-Aware Reselection for Self-Consistency* (CARS), which retains Vanilla SC by default and selectively activates Top-2 reselection under sufficiently strong candidate competition. CARS consists of two stages: competition-aware gating and cluster-level reselection.

3.1 Competition-Aware Gating

Candidate support is used to determine whether auxiliary reselection is needed. CARS measures the competition between the two leading candidates using $R = q_2/q_1$. Given a gating threshold $\rho \in [0, 1]$, CARS retains the Vanilla SC decision C_1 when $R < \rho$. When $R \geq \rho$, the two leading candidates have sufficiently similar support and are passed to the cluster-level reselection stage. Thus, R serves only as a gating variable and does not directly modify the candidate ranking.

This design follows the observation in Section 2.4 that auxiliary reselection is more useful when candidate frequency provides weaker separation between the two leading candidates. Restricting intervention to such cases also avoids unnecessary and potentially harmful overrides when C_1 maintains a clear support advantage.

3.2 Cluster-Level Reselection

Once the gate is activated, the trajectory-level signal is used only to discriminate between C_1 and C_2. CARS compares their within-cluster mean signals $\bar{s}_1$ and $\bar{s}_2$, as defined in Eq. (5). Unless otherwise specified, we use tail-30% certainty as the trajectory-level signal $s(\tau)$.

Combining the two stages, CARS selects the following cluster and returns its answer:

$$\hat{C}_{\mathrm{CARS}} = \begin{cases} C_2, & R \geq \rho \text{ and } \bar{s}_2 > \bar{s}_1, \\ C_1, & \text{otherwise.} \end{cases} \tag{6}$$

CARS therefore deviates from the Vanilla SC decision C_1 only when the competition gate is activated and C_2 has a strictly higher cluster-level signal than C_1. Otherwise, it retains C_1. In particular, equal cluster-level signals preserve the Vanilla SC decision.

CARS reuses the sampled trajectories and the probability information recorded during generation. It therefore requires no additional trajectory generation or model decoding passes; the additional computation is limited to post-hoc signal aggregation, competition-aware gating, and Top-2 comparison at the candidate-selection stage.

4 Experiments

4.1 Experimental Setup

Models. We evaluate CARS on three instruction-tuned models from different model families at comparable scales (7B–8B): Llama3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Yang et al., 2024), and InternLM3-8B-Instruct (InternLM Team, 2025). In addition, we analyze Qwen3-8B (Yang et al., 2025) in Section 4.5 to examine whether the competition-conditioned selection pattern persists on a stronger model.

Benchmarks. We evaluate on six reasoning benchmarks: AIME 2024/2025 (Zhang & Math-AI, 2024; 2025), AMC 2023 (AI-MO, 2024), MATH-500 (Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and GPQA Diamond (Rein et al., 2024). AIME 2024 and AIME 2025 are combined into a single evaluation set. The first five benchmarks focus on mathematical reasoning, while GPQA Diamond evaluates scientific reasoning. We report per-benchmark accuracy and pooled overall accuracy.

Baselines. We compare CARS with three inference strategies. *Vanilla Self-Consistency (SC)* selects the most frequently predicted answer (Wang et al., 2023). *Unconditional Top-2 Reselection (UTRS)* always reselects between the two most-supported answer clusters using their cluster-level trajectory signals, without conditioning on candidate competition. *Trajectory-Signal-Weighted Self-Consistency (TSWS)* weights each trajectory's vote by its trajectory-level signal and selects the answer with the highest aggregated weighted support (Zhou et al., 2025). For the main comparison, UTRS, TSWS, and CARS all use the same tail-30% certainty signal.

Experimental Settings. For each problem, we generate 128 reasoning trajectories from the evaluated model and exclude those whose final answers cannot be parsed. All aggregation methods operate on the same remaining valid trajectories, ensuring an identical generation budget and isolating differences in aggregation and candidate selection. For the main experiments in Section 4.2, CARS uses tail-30% certainty and a fixed competition threshold of $\rho = 0.70$, chosen as an empirical operating point for relatively strong Top-2 competition and applied uniformly across all models and benchmarks without model- or benchmark-specific tuning. Section 4.4 further examines alternative trajectory-level signals and competition thresholds.

4.2 Main Results

Table 1 summarizes the main results. Under the shared 128-trajectory budget, CARS improves overall accuracy over Vanilla SC by 0.86, 0.86, and 1.03 percentage points on Llama3.1-8B-Instruct, Qwen2.5-7B-Instruct, and InternLM3-8B-Instruct, respectively, showing consistent gains across the three evaluated model families.

The comparison with UTRS highlights the role of selective intervention. UTRS uses the same trajectory signal and Top-2 reselection rule as CARS but applies reselection unconditionally. This reduces overall accuracy by 5.10, 4.47, and 3.04 points on the three models. Across all evaluations, UTRS produces 271 *Rescue* cases but 491 *Broken* cases, showing that trajectory evidence can correct SC errors while also introducing many harmful overrides when applied indiscriminately. By restricting reselection to stronger-competition instances, CARS produces 116 *Rescue* and 68 *Broken*

Table 1: Comparison of Vanilla SC, UTRS, TSWS, and CARS across three models and six reasoning benchmarks. Accuracy (%) is reported for each benchmark and over the pooled evaluation set. *Ovr. / Res. / Bro.* denotes the numbers of overrides, incorrect→correct predictions, and correct→incorrect predictions relative to Vanilla SC, respectively.

Method	AIME	AMC	MATH	Minerva	Olympiad	GPQA	Overall Acc.	Ovr. / Res. / Bro.
Llama3.1-8B-Instruct								
Vanilla SC	8.33	50.00	71.80	39.71	34.57	32.83	45.30 (-)	–
UTRS	10.00	45.00	62.00	30.88	31.90	**34.34**	40.19 (-5.10)	690 / 86 / 175
TSWS	8.33	50.00	73.20	39.71	35.01	32.83	45.87 (+0.57)	98 / 12 / 2
CARS	**10.00**	**52.50**	**73.20**	**40.07**	**35.01**	33.84	**46.16 (+0.86)**	268 / 42 / 27
Qwen2.5-7B-Instruct								
Vanilla SC	20.00	62.50	82.40	62.50	49.26	32.83	58.26 (-)	–
UTRS	18.33	57.50	76.60	54.04	44.66	**36.87**	53.78 (-4.47)	588 / 85 / 163
TSWS	20.00	62.50	82.60	61.76	49.26	33.33	58.26 (+0.00)	30 / 4 / 4
CARS	**20.00**	**65.00**	**82.60**	**63.24**	**50.15**	35.35	**59.12 (+0.86)**	175 / 33 / 18
InternLM3-8B-Instruct								
Vanilla SC	16.67	57.50	81.20	48.53	48.81	36.87	55.79 (-)	–
UTRS	16.67	57.50	77.20	43.38	46.14	36.36	52.75 (-3.04)	553 / 100 / 153
TSWS	16.67	**57.50**	81.60	48.53	49.11	36.87	56.02 (+0.23)	38 / 5 / 1
CARS	**16.67**	55.00	**81.60**	**48.90**	**50.74**	**38.38**	**56.82 (+1.03)**	183 / 41 / 23

cases, yielding a positive overall accuracy gain on all three models. This contrast shows that the effectiveness of trajectory-based reselection depends strongly on when it is applied.

TSWS provides a complementary comparison by combining candidate frequency and trajectory evidence through signal-weighted voting. It improves over Vanilla SC by 0.57, 0.00, and 0.23 points on the three models, compared with 0.86, 0.86, and 1.03 points for CARS. At the benchmark level, CARS matches or exceeds TSWS in 17 of the 18 model–benchmark settings. TSWS is more conservative, with 166 overrides, 21 *Rescue*, and 7 *Broken* cases, versus 626, 116, and 68 for CARS. Thus, weighted voting causes fewer harmful overrides but also corrects fewer erroneous SC decisions.

Overall, these results support the two CARS design choices: candidate competition determines when trajectory evidence should influence the SC decision, while the trajectory signal remains separate from candidate frequency and is used to distinguish the leading candidates once reselection activates.

4.3 Competition-Aligned Gains Across Models and Benchmarks

Figure 5 examines whether the competition-dependent gain pattern identified in Section 2.4 persists across individual models and benchmarks. Across many of the displayed model–benchmark pairs, positive net contributions concentrate in regions of stronger Top-2 competition. The recurrence of this pattern across individual settings shows that the competition dependence observed in the pooled analysis is not merely an artifact of aggregating heterogeneous model–benchmark pairs.

At the same time, the positive-gain region varies across settings. In some model–benchmark pairs, reselection becomes beneficial at moderate competition, whereas in others gains appear only at higher R. Thus, the favorable competition level is not universal. Figure 5 supports a competition-aligned view: beneficial regions tend to emerge as the leading candidates become more competitive, without implying a single optimal threshold across models and benchmarks.

4.4 Sensitivity Analysis

We examine the sensitivity of CARS to the trajectory-level signal and the competition threshold.

Signal sensitivity. We fix the competition threshold at $\rho = 0.70$ and compare four trajectory signals: full confidence, tail-30% confidence, full certainty, and tail-30% certainty. As shown in Table 2, all variants improve over Vanilla SC across all three models, with varying gains. Tail-based signals consistently match or outperform their full-trajectory counterparts, and tail-30% certainty achieves the

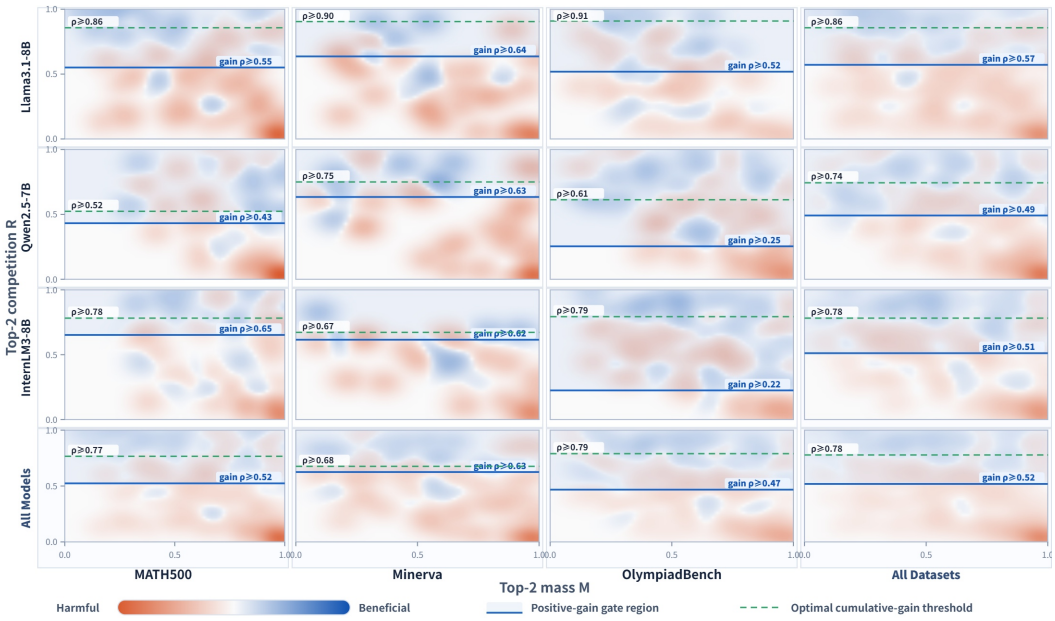


Figure 5: **Signed net-gain landscapes across models and benchmarks.** Normalized net gain of certainty-based Top-2 reselection over Vanilla SC across (M, R). Blue/red indicate beneficial/harmful regions; solid/dashed lines denote positive-gain and optimal cumulative-gain thresholds. The last row and column show aggregated results.

largest average gain. These results show that CARS generalizes across trajectory signals, although performance depends on the signal choice.

Threshold sensitivity. We fix tail-30% certainty and vary the competition threshold from $\rho = 0.50$ to 0.90 with a step size of 0.01. As shown in Figure 6, the threshold yielding the largest gain varies across models, but stronger improvements consistently occur within a moderately high competition range. Lower thresholds trigger reselection on more instances but also introduce more harmful overrides, whereas higher thresholds make the intervention more selective at the cost of fewer opportunities for correction. Intervention statistics are provided in Appendix B.2. Overall, CARS does not rely on a narrowly tuned threshold, but benefits from restricting trajectory-based reselection to sufficiently competitive cases.

Table 2: Overall accuracy (%) with different trajectory signals at $\rho = 0.70$.

Method	Llama3.1	Qwen2.5	InternLM3
Vanilla SC	45.30	58.26	55.79
Conf. (Full)	45.70 (+0.40)	58.66 (+0.40)	56.25 (+0.46)
Conf. (T30%)	45.70 (+0.40)	59.12 (+0.86)	56.54 (+0.75)
Cert. (Full)	46.16 (+0.86)	58.83 (+0.57)	56.31 (+0.52)
Cert. (T30%)	46.16 (+0.86)	59.12 (+0.86)	56.82 (+1.03)

4.5 Competition-Conditioned Selection on a Stronger Model

We additionally analyze Qwen3-8B to examine whether the competition-dependent selection pattern persists on a stronger model. As in Section 2.4, we restrict the analysis to the Top-2-controlled subset to isolate candidate selection from candidate coverage. At each competition level $R = q_2/q_1$, we compare Vanilla SC, which selects C_1, with certainty-based Top-2 selection using cluster-mean tail-30% certainty. As shown in Figure 7, Vanilla SC accuracy generally decreases as R increases, reproducing the competition-dependent degradation observed in Section 2.4. Vanilla SC remains more accurate across much of the range, but the gap narrows at high R, where certainty-based selection approaches or exceeds it in some bins.

Compared with the earlier models, the complementary value of trajectory certainty on Qwen3-8B is more concentrated in the high-competition regime. This indicates that candidate competition remains informative about when trajectory evidence becomes useful, while the competition level

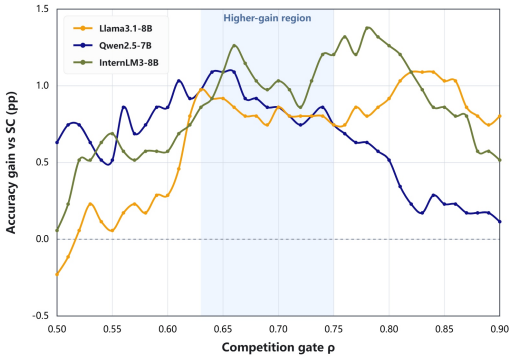


Figure 6: Accuracy gain across competition thresholds ρ using tail-30% certainty.

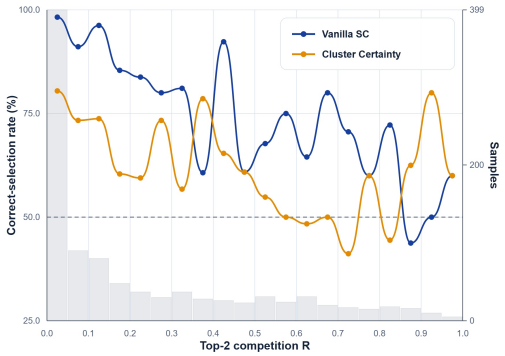


Figure 7: Qwen3-8B selection accuracy across Top-2 competition levels.

at which reselection becomes beneficial varies across models. Because the highest-R bins contain fewer samples, the crossover should be interpreted descriptively rather than as a stable threshold.

5 Related Work

Self-Consistency and Test-Time Answer Aggregation. Self-consistency (SC) improves the reasoning ability of large language models by sampling multiple reasoning trajectories and selecting the most frequent final answer (Wang et al., 2023). Subsequent studies extend this paradigm through adaptive sampling, early stopping, test-time compute allocation, and alternative aggregation strategies (Aggarwal et al., 2023; Li et al., 2024; Snell et al., 2025; Chen et al., 2024; Kim et al., 2026). These methods generally rely on agreement among sampled answers, with candidate support serving as an important indicator of selection reliability. Recent work further examines vote margins, candidate support, and answer-distribution structure to characterize when consensus-based selection is reliable (Feng et al., 2026; Li et al., 2025; Cordero-Encinar & Duncan, 2025). Motivated by these studies, we analyze SC reliability from the perspective of candidate competition, focusing in particular on the relative support of the Top-2 answer candidates. From this perspective, we use the degree of Top-2 competition as a criterion for determining when discriminative evidence is needed.

Trajectory-Level Signals for Reasoning Aggregation. Another line of work leverages information contained within reasoning trajectories to assess their quality. Common signals include sequence likelihood, token-level confidence, predictive entropy, certainty, and other probability-based uncertainty measures (Lin et al., 2024; Kang et al., 2025; Fu et al., 2026; Liu et al., 2026). Beyond model-native probability signals, prior work has also employed learned verifiers, reward models, process-level evaluators, and latent-space representations to score or compare reasoning trajectories (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024; Xie et al., 2024). These signals are commonly incorporated through Best-of-N selection, reranking, weighted voting, or verifier-guided aggregation, providing evidence beyond answer frequency. Existing studies primarily focus on identifying effective trajectory-level signals or developing better ways to integrate them into the aggregation process. However, trajectory-level signals can be noisy, and incorporating them indiscriminately may interfere with otherwise reliable frequency-based decisions (Zenn & Geiping, 2026). Motivated by this consideration, our work treats candidate support and trajectory-level signals as complementary yet distinct sources of evidence, selectively introducing trajectory-level signals only when the leading candidates are sufficiently competitive.

6 CONCLUSION

In this work, we introduced CARS, a competition-aware reselection framework that retains frequency-based selection under weak competition and uses trajectory evidence to distinguish the Top-2 candidates under strong competition. Our results highlight candidate competition as a practical criterion for determining when trajectory-level signals should influence self-consistency.

AI USE STATEMENT

During the preparation of this manuscript, AI tools (e.g., OpenAI's ChatGPT) were used solely to assist with language polishing, clarity, and keyword selection. The authors take full responsibility for all concepts, experimental designs, analyses, and conclusions. All references were independently verified, and no confidential or unpublished data were shared.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12375–12396. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.761. URL https://doi.org/10.18653/v1/2023.emnlp-main.761.

AI-MO. Amc 2023, 2024. URL https://huggingface.co/datasets/AI-MO/aimo-validation-amc.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language models. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://openreview.net/forum?id=LjsjHF7nAN.

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. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Paula Cordero-Encinar and Andrew B. Duncan. Certified self-consistency: Statistical guarantees and test-time training for reliable reasoning in llms. *CoRR*, abs/2510.17472, 2025. doi: 10.48550/ARXIV.2510.17472. URL https://doi.org/10.48550/arXiv.2510.17472.

Austin Feng, Marius Alonso, Ambroise Odonnat, Vasilii Feofanov, and Ievgen Redko. Optimal self-consistency for efficient reasoning with large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=sUTC7JE781.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8LqHs0KIM7.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

InternLM Team. Internlm3-8b-instruct. https://huggingface.co/internlm/internlm3-8b-instruct, 2025. Model card.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza

Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/1c7eff166a8e345f664f0faa8f4e4d2e-Abstract-Conference.html.

Junseok Kim, Nakyeong Yang, Kyungmin Min, and Kyomin Jung. Reliability-aware adaptive self-consistency for efficient sampling in LLM reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 21575–21590. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.1085. URL https://doi.org/10.18653/v1/2026.findings-acl.1085.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ndR8Ytrzhh.

Yiwei Li, Ji Zhang, Shaoxiong Feng, Peiwen Yuan, Xinglin Wang, Jiayi Shi, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Revisiting self-consistency from dynamic distributional alignment perspective on answer aggregation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 25208–25223. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.1293. URL https://doi.org/10.18653/v1/2025.findings-acl.1293.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 10351–10368. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.578. URL https://doi.org/10.18653/v1/2024.emnlp-main.578.

Shuo Liu, Ding Liu, and Shi-Ju Ran. Confidence geometry reveals trace-level correctness in large language model reasoning. *CoRR*, abs/2605.16824, 2026. doi: 10.48550/ARXIV.2605.16824. URL https://doi.org/10.48550/arXiv.2605.16824.

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

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

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.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275, 2022. URL https://doi.org/10.48550/arXiv.2211.14275.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Zhihui Xie, Jizhou Guo, Tong Yu, and Shuai Li. Calibrating reasoning in language models with internal consistency. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/d037fd021c9aace128b8ce25001cdb6c-Abstract-Conference.html.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

Johannes Zenn and Jonas Geiping. When are likely answers right? on sequence probability and correctness in LLMs. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026. URL https://openreview.net/forum?id=mKxTWhajn5.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=E0PaeSszLz.

A Experimental Details and Answer Evaluation

A.1 Inference and Hyperparameter Settings

All experiments use vLLM for inference without additional training or fine-tuning. For each problem, we sample 128 reasoning trajectories with temperature 0.7, top-$p = 0.8$, a maximum generation length of 8,192 tokens, and a fixed random seed of 42. Confidence uses the generated token's log probability; certainty uses the Top-20 token log probabilities, with top-K size $K = 20$. Both use the sampling distribution after temperature and top-p; the Top-20 subset is not separately renormalized. Here T is the number of generated tokens used to score each trajectory, and logarithms are natural. Tail confidence applies Eq. (2) to the final $m = \lceil 0.3T \rceil$ tokens, exponentiating before cluster averaging. Each model uses its tokenizer-provided chat template, with explicit thinking mode disabled. Experiments are run on eight NVIDIA RTX 5880 GPUs, with each 7B–8B model loaded on a single GPU.

A.2 Datasets

We evaluate all methods on a collection of mathematical and scientific reasoning benchmarks with different domains and difficulty levels. The mathematical benchmarks include AIME 2024, AIME 2025, AMC 2023, MATH-500, Minerva Math, and OlympiadBench, while GPQA Diamond provides an evaluation of scientific reasoning. AIME 2024 and AIME 2025 are combined into a single evaluation set in the main experiments. Table 3 summarizes the number of problems used from each benchmark.

Table 3: Number of evaluation problems used for each benchmark. AIME 2024 and AIME 2025 are combined into a single AIME evaluation set in the main experiments.

Dataset	Number of Problems
AIME 2024	30
AIME 2025	30
AMC 2023	40
MATH-500	500
Minerva Math	272
OlympiadBench	674
GPQA Diamond	198

AIME 2024/2025. The American Invitational Mathematics Examination (AIME) consists of challenging competition-style mathematics problems that typically require multi-step reasoning across topics such as algebra, geometry, number theory, and combinatorics. We use all 30 problems from AIME 2024 and all 30 problems from AIME 2025, and combine them into a single 60-problem evaluation set in our experiments.

AMC 2023. AMC 2023 is a competition-style mathematics benchmark covering a broad range of high-school mathematical topics. Compared with AIME, the problems generally involve shorter reasoning chains while still requiring nontrivial arithmetic, algebraic, geometric, and combinatorial reasoning. We evaluate on 40 problems from AMC 2023.

MATH-500. MATH-500 is a 500-problem subset of the MATH benchmark designed to evaluate advanced mathematical problem solving. It covers a diverse range of mathematical domains, including algebra, geometry, number theory, counting and probability, precalculus, and calculus. We use all 500 problems in MATH-500.

Minerva Math. Minerva Math contains quantitative reasoning problems spanning a range of mathematical domains and difficulty levels. It provides a complementary setting to competition-oriented benchmarks by including problems with diverse formats and reasoning requirements. We evaluate on 272 problems.

OlympiadBench. OlympiadBench is an olympiad-level reasoning benchmark containing challenging problems that often require longer reasoning chains and nontrivial problem-solving strategies. We use 674 mathematical reasoning problems from OlympiadBench in our evaluation.

GPQA Diamond. GPQA Diamond is a challenging subset of GPQA designed to evaluate graduate-level scientific reasoning. The questions span multiple scientific domains and typically require both specialized knowledge and multi-step reasoning. We evaluate on all 198 problems in the GPQA Diamond subset.

A.3 Prompt Templates

We use a shared prompt template for all mathematical reasoning benchmarks and a separate multiple-choice template for GPQA Diamond. The placeholder `{instruction}` is replaced by the original problem statement. All prompts are passed through the corresponding tokenizer-provided chat template before generation.

Mathematical Reasoning Benchmarks. For AIME 2024/2025, AMC 2023, MATH-500, Minerva Math, and OlympiadBench, we use the following prompt:

```
Question:
{instruction}
Please reason step by step, and put your final answer
within \boxed{}.
```

GPQA Diamond. For GPQA Diamond, we use the following multiple-choice prompt:

```
Question:
{instruction}

Please reason step by step, then select exactly one
option from A, B, C, or D.
Put only the letter of your final selected answer (A, B,
C, or D) within \boxed{}.
Do not include the option text or any other content
inside the box.
```

A.4 Multi-Stage Answer Evaluation

Evaluating the correctness of generated answers is nontrivial because the same correct answer may appear in multiple surface forms. For example, mathematically equivalent expressions may differ in formatting, simplification, notation, or the way intermediate quantities are represented. Consequently, relying only on exact string matching can incorrectly classify semantically equivalent answers as different. At the same time, relying entirely on an LLM-based verifier may introduce additional uncertainty into the evaluation process. We therefore use a multi-stage answer evaluation procedure that combines deterministic matching, LLM-assisted assessment, and manual verification.

Inference clusters trajectories solely by their parsed final-answer strings; equivalence checks and reference answers are used only to evaluate correctness, never to form voting clusters or select an answer. Thus, different strings remain separate voting clusters even when both are correct.

The evaluation begins with deterministic matching based on string normalization and a collection of task-dependent equivalence rules. Generated answers that can be verified as equivalent to the reference answer at this stage are directly assigned to the *correct equivalence class*. This stage handles answer forms that can be reliably resolved without introducing an additional model-based judgment.

Answers that cannot be resolved by deterministic matching are passed to a second stage for LLM-assisted preliminary assessment and manual verification. This stage is used to determine whether an unmatched answer is nevertheless semantically or mathematically equivalent to the reference answer. If the answer is verified as correct, it is recorded as a correct equivalent answer; otherwise, it is labeled incorrect. These labels do not merge or change the string-based voting clusters.

The resulting evaluation judgments are reused as more generated trajectories are scored. Once a particular answer form or an equivalent variant has been verified and incorporated into an existing equivalence class, subsequent matching answers can be resolved directly through the deterministic stage. As the number of evaluated samples increases, the set of known equivalent answer forms therefore becomes progressively more complete, reducing the fraction of cases that require LLM-assisted assessment or manual verification. Reference-based Top-2 filtering is used only for diagnostic analyses, not inference.

Aggregation conventions. Each cluster is a multiset of sampled records: repeated samples each count as a vote, and support is $q_i = |C_i| / \sum_j |C_j|$. Support ties retain the common SC ordering across methods, so $|C_1| \geq |C_2| \geq \cdots$. With only one cluster, its answer is returned without reselection; with no valid trajectories, no cluster can be selected. For at least two clusters, $\rho = 0$ recovers UTRS, whereas $\rho = 1$ still permits reselection when supports tie.

Signal-weighted voting. TSWS selects a cluster by

$$\hat{C}_{\mathrm{TSWS}} = \arg\max_{C_i} \sum_{\tau \in C_i} s(\tau) = \arg\max_{C_i} |C_i| \bar{s}_i,$$

and returns its answer. The main experiments sum raw tail-30% certainty, without exponentiation or z-score normalization. For confidence weights, Eq. (2) already exponentiates each trajectory's mean log probability before summation.

B Additional Analysis of Trajectory Signals and Reselection

B.1 Within-Question Discrimination of Trajectory Signals

To better understand the correctness-related information captured by trajectory-level probability signals, we examine their ability to discriminate between correct and incorrect outcomes within the same question. We consider both confidence and certainty, as well as local variants computed over different portions of the trajectory. In particular, we compare full-trajectory confidence and certainty with tail-based signals computed from the final portion of the reasoning trajectory.

Table 4 reports within-question AUROC (area under the receiver operating characteristic curve) for the three main models; higher scores indicate better separation of correct from incorrect outcomes. For confidence, we compare the full-trajectory signal with its tail-30% variant. For certainty, we additionally vary the local window from the final 10% to 40% of the trajectory to examine how the location and extent of local probability information affect discrimination.

Table 4: Within-question AUROC of different trajectory-level probability signals. We compare full-trajectory and tail-based confidence and certainty on the three models used in our main experiments. The last column reports the average across the three models. The best result in each column is shown in bold.

Signal	Llama3.1-8B	Qwen2.5-7B	InternLM3-8B	Average
Full Confidence	0.5570	0.5901	0.5944	0.5805
Tail-30% Confidence	0.5687	0.5923	0.5899	0.5836
Full Certainty	0.5927	0.6086	0.6063	0.6025
Tail-10% Certainty	0.5928	0.5980	**0.6243**	0.6050
Tail-20% Certainty	0.6012	0.6110	0.6229	0.6117
Tail-30% Certainty	**0.6054**	0.6190	0.6210	**0.6151**
Tail-40% Certainty	0.6036	**0.6192**	0.6195	0.6141

B.2 Intervention Statistics across Competition Thresholds

To further examine how the competition threshold affects the behavior of CARS, we report detailed intervention statistics at representative thresholds $\rho \in \{0.50, 0.60, 0.70, 0.80, 0.90\}$. For each threshold, we report the number of predictions overridden relative to Vanilla SC, together with the

Table 5: Intervention statistics of CARS under different competition thresholds ρ using tail-30% certainty. Overrides denotes the number of predictions changed relative to Vanilla SC; Rescue and Broken denote incorrect→correct and correct→incorrect changes, respectively. Net Rescue is Rescue minus Broken, and Accuracy Gain reports the absolute accuracy change over Vanilla SC in percentage points.

Model	ρ	Overrides	Rescue	Broken	Net Rescue	Acc. Gain (pp)
Llama3.1-8B	0.50	443	58	62	-4	-0.23
Llama3.1-8B	0.60	355	51	46	$+5$	$+0.29$
Llama3.1-8B	0.70	268	42	27	$+15$	$+0.86$
Llama3.1-8B	0.80	200	34	18	$+16$	$+0.92$
Llama3.1-8B	0.90	107	19	5	$+14$	$+0.80$
Qwen2.5-7B	0.50	276	54	43	$+11$	$+0.63$
Qwen2.5-7B	0.60	222	44	29	$+15$	$+0.86$
Qwen2.5-7B	0.70	175	33	18	$+15$	$+0.86$
Qwen2.5-7B	0.80	126	23	14	$+9$	$+0.52$
Qwen2.5-7B	0.90	59	8	6	$+2$	$+0.11$
InternLM3-8B	0.50	304	58	57	$+1$	$+0.06$
InternLM3-8B	0.60	244	50	40	$+10$	$+0.57$
InternLM3-8B	0.70	183	41	23	$+18$	$+1.03$
InternLM3-8B	0.80	115	31	9	$+22$	$+1.26$
InternLM3-8B	0.90	53	13	4	$+9$	$+0.52$

numbers of *Rescue* cases (incorrect→correct) and *Broken* cases (correct→incorrect). Net Rescue is defined as the difference between Rescue and Broken cases. The corresponding accuracy gain over Vanilla SC is also reported in Table 5.

As the competition threshold increases, CARS intervenes on progressively fewer instances, reflecting a more selective use of trajectory-based reselection. At lower thresholds, the larger number of interventions creates more opportunities for Rescue, but also introduces more Broken cases. For example, at $\rho = 0.50$, Llama3.1-8B produces 58 Rescue cases but 62 Broken cases, resulting in a negative net effect. Restricting reselection to more competitive instances substantially reduces harmful overrides.

The threshold yielding the strongest net improvement differs across models. Llama3.1-8B and InternLM3-8B remain effective at relatively high thresholds, whereas Qwen2.5-7B obtains its strongest gains at moderate competition levels. Nevertheless, $\rho = 0.70$ yields a positive Net Rescue and accuracy improvement for all three models, illustrating that the benefit of reselection is generally concentrated in sufficiently competitive cases rather than under unrestricted intervention.

C Construction of Analysis Figures

C.1 Vanilla SC Accuracy over the Candidate-Structure Space

Figure 1. Figure 1 shows the local Vanilla SC accuracy over the (M, R) plane. Instances are first assigned to a two-dimensional grid, after which the numbers of correct predictions and total instances are smoothed separately. Their ratio gives the displayed local accuracy. Regions with insufficient local support are left unshown. Panel (a) uses all instances from the three main models, while panel (b) uses only instances whose correct answer is among the Top-2 candidates.

C.2 Signed Net Effect of Top-2 Reselection

Figure 3. For each instance, we define

$$d_i = \mathbb{I}(\text{reselection correct}) - \mathbb{I}(\text{Vanilla SC correct}),$$

so that Rescue and Broken contribute $+1$ and -1; all remaining cases contribute 0, including answer changes that preserve correctness. Figure 3 smooths these signed changes over the (M, R) plane. The displayed value corresponds to a smoothed $N_{\text{rescue}} - N_{\text{broken}}$, rather than a local accuracy

difference or percentage. The figure pools instance-level results from the three main models and all six reported benchmarks, restricted to the Top-2-controlled subset; color intensity is scaled by the maximum absolute smoothed value over the figure.

C.3 Model- and Benchmark-Specific Gain Landscapes

Figure 5. Figure 5 uses the same signed-effect construction, but reports results separately for models and the three displayed benchmarks: MATH-500, Minerva Math, and OlympiadBench. Each panel is normalized by the total number of problems in that panel, rather than the local sum of smoothing weights. The aggregate row and column use instance-weighted aggregation; therefore, the "All Datasets" column refers only to the three displayed benchmarks.

C.4 Within-Question Standardization of Cluster Signals

For Figure 2, each question's J string-based clusters are first assigned the mean tail-30% certainty $\bar{s}_i$ from Eq. (5). We then apply an unweighted population z-score across all clusters of that question:

$$\mu = \frac{1}{J}\sum_{i=1}^{J}\bar{s}_i, \qquad \sigma = \sqrt{\frac{1}{J}\sum_{i=1}^{J}(\bar{s}_i-\mu)^2}, \qquad z_i = \frac{\bar{s}_i-\mu}{\sigma} \quad (\sigma > 0).$$

Each cluster has equal weight regardless of support. Zero denotes the question's mean cluster score; positive and negative values denote scores above and below that mean. The correct cluster is the highest-support correct cluster; incorrect clusters are ranked by support as Wrong rank-1, rank-2, and so on. This standardization is only for visualization; aggregation uses unstandardized signals.