

Structural Signatures of Correct Mathematical Reasoning Traces

Maria Marina^{1,2} and Alexander Panchenko^{1,2}

¹AIRI, ²Skoltech

Correspondence: {m.marina.scientia}@gmail.com

Abstract

Self-consistency improves reasoning by voting over sampled final answers, but the vote discards how those answers were reached. We ask whether the internal organization of traces provides an interpretable signal that frequency misses. For each candidate answer, we cluster recurring intermediate statements into semantic *checkpoints*, retain checkpoint orders that are consistent across traces, and identify relations that appear both directly and with intermediate checkpoints inserted. These relations form answer-local shortcut graphs. Our score is the number of graph edges beyond a spanning tree in the most cross-linked component, normalized by answer frequency and smoothed over nearby sampling budgets. The intuition is deliberately structural: correct traces may agree on core dependencies while varying in local decomposition, whereas a wrong majority can arise from repeated copies of one failure route. On MATH500, the structural selector is mostly neutral when evidence is scarce, but after sufficient sampling it frequently repairs majority errors, reaching up to a 2-point accuracy gain over majority voting.

1 Introduction

Chain-of-thought prompting exposes intermediate reasoning (Wei et al., 2022), and self-consistency improves reliability by sampling multiple traces and returning the modal answer (Wang et al., 2023). Repeated sampling has consequently become a standard form of test-time scaling (Brown et al., 2024; Snell et al., 2025). However, the standard aggregation step – majority voting — reduces each rich reasoning trace to a single categorical vote, discarding information about how the answer was reached. Ten near-duplicate derivations and ten solutions that recover the same mathematical dependencies through different local routes contribute exactly the same support.

We study the information lost by this compression. Our hypothesis is not simply that correct traces are “more diverse.” Useful diversity must coexist with *consistency*: traces should preserve recurrent mathematical states and their relative order, while differing in which intermediate states are made explicit. This distinction motivates our notion of a *shortcut*. Suppose several traces agree that state A must precede state B . One trace may realize the dependency directly, $A \rightarrow B$, whereas another makes an intermediate argument explicit, $A \rightarrow D \rightarrow B$. Both traces preserve the same dependency, but the second expands a step that the first takes as a shortcut. We regard a relation as especially informative when both realizations recur across traces: it is stable enough to survive different derivations, yet flexible enough not to be a literal copy of one route. When several such relations overlap, the answer-local graph becomes cross-linked rather than tree-like.

We therefore seek a selector whose decision can be traced back to this observable organization of reasoning. Figure 1 visualizes the central pattern. In the first two repair cases, the correct structural candidate has more excess connectivity relative to its frequency than the wrong majority despite receiving fewer votes; the final pair shows the corresponding failure mode, where a small change in the shortcut graph makes a wrong answer structurally preferable.

Our contribution is a simple answer-local construction that combines *checkpoint consistency* with *route diversity*. We (i) build semantic checkpoint communities across traces, (ii) retain precedence relations that are stable and survive both direct and expanded realizations, and (iii) score the most redundantly connected component by its excess edges per supporting trace. The resulting choice can be inspected as a graph of the evidence that caused it.

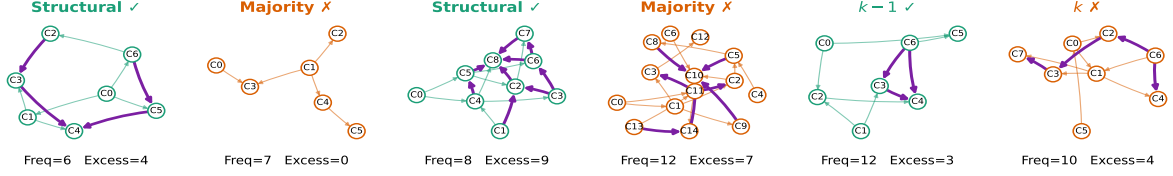


Figure 1: Representative shortcut graphs. The first two pairs are majority-error repairs: the structural selector chooses the correct answer although it is less frequent. In the early case, the structural candidate has frequency 6 and 4 excess edges, while the majority has frequency 7 and zero excess; in the later case the comparison is 8/9 versus 12/7 (frequency/excess). The final pair is a structural-drift failure from $k - 1$ to k . Thick purple edges show one spanning-tree decomposition of excess connectivity; the number of excess edges is invariant although the particular highlighted edges depend on the chosen spanning tree.

2 Related Work

Self-consistency samples diverse chains but aggregates only final answers (Wang et al., 2023). Universal self-consistency and semantic uncertainty improve answer-level grouping by clustering outputs by meaning (Chen et al., 2023; Kuhn et al., 2023). Learned verifiers and process supervision instead score complete solutions or individual steps (Cobbe et al., 2021; Lightman et al., 2024), while RISC ranks answer candidates using frequency, semantic centrality, and trace-level features (Marina et al., 2026). We focus on a different signal: the *within-answer organization of intermediate reasoning*. Final-answer normalization is fixed, and a fixed semantic matcher exposes recurrent checkpoints whose ordering and connectivity can be inspected directly.

3 Structural Signature of an Answer

3.1 Semantic checkpoints

Intuition. We first look for reasoning steps that play the same semantic role in different traces ending in the *same* answer. These recurring states are our checkpoints. The construction is answer-local because two traces that reach different final answers may reuse similar-looking algebra while supporting different candidate solutions; we therefore never use cross-answer matches to create evidence for a candidate.

For question q , answer a , and set of k sampled traces, let $\mathcal{T}_{q,a}^{(k)}$ contain the traces ending in a , with frequency $n_{q,a}^{(k)}$. We split each trace into ordered reasoning units after removing the final-answer suffix. Every cross-trace pair of units is scored in both directions by Qwen3-Reranker-0.6B (Zhang et al., 2025); the pair score is the minimum directional raw logit. Model’s joint pairwise encoding is well suited to fine-grained semantic matching, including

mathematical text that mixes natural language and symbolic expressions. Since the score depends on input order, we evaluate both $s(u, v)$ and $s(v, u)$ and use the minimum raw logit as a conservative symmetric similarity score. Within each answer family, we use the empirical 0.90 quantile as the similarity threshold.

We greedily merge mutually similar units using complete-link similarity, while forbidding two units from the same trace in one community. A community is retained only if it occurs in at least 5 distinct traces. Each retained community C_i is a recurring semantic checkpoint. Finally, every original trace is projected back onto its matched communities in trace order. Thus a free-form derivation becomes an ordered checkpoint sequence such as $\pi_t = (C_0, C_2, C_5)$ or $\pi_{t'} = (C_0, C_1, C_2, C_5)$. The next stage compares these discrete sequences rather than the raw text.

3.2 Stable relations with variable local detail

For checkpoints C_i, C_j , let F_{ij} and R_{ij} count traces in which they co-occur in the two possible orders. We require support $F_{ij} + R_{ij} \geq 3$ and define ordering consistency

$$\rho_{ij} = \frac{\max(F_{ij}, R_{ij})}{F_{ij} + R_{ij}}. \quad (1)$$

The larger count determines the dominant direction. For example, if C_0 precedes C_2 in most traces, then the candidate edge is $C_0 \rightarrow C_2$.

An edge is inserted in the shortcut graph *only if three conditions hold*: (1) the pair co-occurs in at least 3 traces; (2) its dominant order has consistency $\rho_{ij} \geq 0.80$; and (3) that same order is observed both *adjacently* and *non-adjacently*. Concretely, sequences (C_0, C_2) and (C_0, C_1, C_2) provide the two required realizations of $C_0 \rightarrow C_2$: the first uses the relation directly, while the second

inserts checkpoint C_1 between its endpoints. A pair seen only as (C_0, C_2) is repeated but not structurally varied; a pair seen in both orders is varied but not consistent. We keep only the intersection of consistency and local variation. We call the resulting relations *shortcut edges* because one trace can traverse a stable dependency directly while another expands it into several steps.

3.3 Excess connectivity

Let $G_{q,a}^{(k)} = (V, E)$ be the directed shortcut graph for answer a , where V is the set of retained checkpoint communities and E is the set of shortcut edges satisfying the three conditions above. The graph need not be fully connected, so we examine its weakly connected components (WCCs): maximal node sets that are connected after edge directions are temporarily ignored. Direction is used to decide whether a shortcut relation is valid; it is ignored only for the following topological count.

For a WCC H with vertex set V_H and edge set E_H , define

$$\mu(H) = [|E_H| - |V_H| + 1]_+. \quad (2)$$

For a connected undirected graph, $|V_H| - 1$ edges are sufficient to form a spanning tree. Hence $\mu(H)$ counts exactly how many edges remain beyond that minimum: zero for a tree, one for a component with one independent extra connection, and larger values for increasingly cross-linked structure. We call these *excess edges*. They are the quantity highlighted in purple in Figure 1; the particular spanning tree used for visualization is arbitrary, but the count $\mu(H)$ is invariant.

We keep only the most cross-linked component of an answer and normalize its excess by the number of traces supporting that answer:

$$x_{q,a}^{(k)} = \frac{\max_{H \in \text{WCC}(G_{q,a}^{(k)})} \mu(H)}{\max(n_{q,a}^{(k)}, 1)}. \quad (3)$$

We call this the *largest-component excess-edges-per-trace* score. The normalization matters because a frequent answer has more opportunities to accumulate edges merely by having more traces. At inference time we smooth $x_{q,a}^{(k)}$ with a trailing 5-prefix mean, $\bar{x}_{q,a}^{(k)}$, and choose the answer with the largest smoothed score among candidates observed at least 2 times. Ties are broken by frequency; consequently, if every candidate has zero structural evidence, the procedure reduces to majority voting.

The metric is therefore not a generic connectedness score. Stable checkpoint order supplies *consistency*; requiring both adjacent and non-adjacent realizations establishes that the dependency survives different local decompositions; and excess edges measure how many of those stable shortcuts overlap inside one connected reasoning structure. A repeated failure template can be frequent yet remain tree-like, while correct traces can reconstruct the same dependencies through several compatible routes.

4 Experimental Setup

We use 400 questions from the MATH500 trace corpus training split as a development set for fixing hyperparameters and analyzing their sensitivity (Section 6), while the test split is reserved for final evaluation. MATH500 contains 500 competition-level mathematics problems derived from MATH (Hendrycks et al., 2021; Lightman et al., 2024). We use Meta-Llama-3.1-8B-Instruct (Llama Team, 2024), which generates 99 zero-shot reasoning traces per problem (temperature 0.9, top_p=1.0), for 49,500 traces in total. We evaluate nested prefixes $k = 5, \dots, 99$ against majority voting on the same first k traces. The exact generation and checkpoint-equivalence prompts are provided in Appendix A.

We use a single budget-independent configuration throughout the final evaluation: community support 5, precedence support 3, ordering consistency 0.80, similarity quantile 0.90, minimum answer frequency 2, and rolling window 5. These values are fixed globally rather than tuned separately for individual sampling budgets. Section 6 reports one-factor sensitivity around this reference configuration. We made the code publicly available.¹

5 Results and Analysis

When does structure help? Figure 2 shows the rolling structural selector across all sampling budgets. With very small prefixes it mostly coincides with majority voting because too few traces support recurrent checkpoint structure. Once enough evidence accumulates, the structural signal becomes useful: the net advantage is positive across most later prefixes, peaks at **10 additional correct questions** (a 2-point accuracy gain on 500 problems),

¹<https://github.com/zlatamaria/structural-signatures>

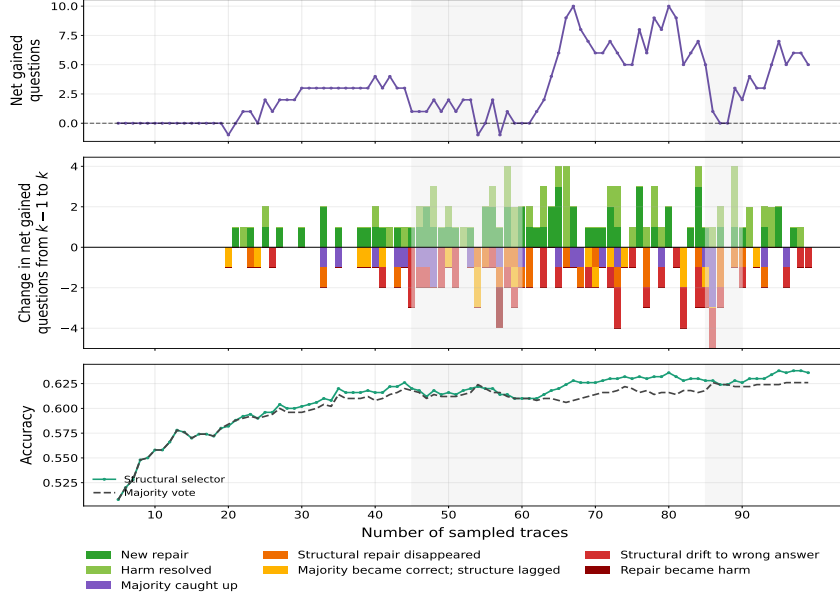


Figure 2: Performance of the rolling structural selector over nested sample sizes. Top: net additional correct questions relative to majority voting. Middle: six question-level transition types that increase or decrease this advantage from one prefix to the next; identical downward changes can reflect baseline catch-up or genuine structural failure. Bottom: matched accuracy of the structural selector and majority. Shaded regions mark the two visible valleys discussed in the text.

and finishes at **+5 questions** at $k = 99$. The bottom panel shows the corresponding accuracy curves: the question-level gains in the upper panels translate into a small but recurrent advantage over majority voting, particularly at larger sample sizes.

Why does the advantage fluctuate? For each question, the structural selector contributes $+1$ relative to majority for a *repair* (majority wrong, structure correct), -1 for a *harm* (majority correct, structure wrong), and 0 otherwise. The middle panel decomposes how this contribution changes from $k-1$ to k into six events. *New repair* is $0 \rightarrow +1$ and *harm resolved* is $-1 \rightarrow 0$. A lost advantage $+1 \rightarrow 0$ has two very different explanations: in *majority caught up*, structure remains correct and majority moves to the same answer; in *structural repair disappeared*, majority remains wrong and structure loses its correction. Likewise, $0 \rightarrow -1$ can mean that *majority became correct; structure lagged*, or a genuine *structural drift to wrong answer* while majority stays correct.

The visible valleys are therefore not simply periods in which structural evidence disappears. Around $k = 45-60$, the number of questions with no structural evidence decreases from 227 to 173 and the number of metric ties from 206 to 158, while the mean score margin increases. The dip is driven by a handful of question-level state changes

rather than by a global collapse of the graph signal. The late valley is even more illustrative: from $k = 85$ to 86 , net gain drops by five questions, but three of the four lost repairs occur because majority changes to the answer that the structural selector was already choosing. In other words, part of the apparent loss is *majority recovery* as additional samples arrive. The remaining new harms are genuine failures in which the shortcut topology changes enough to favor a wrong candidate. This mixture explains why the net-gain curve can be jagged even as aggregate structural evidence becomes denser with larger prefixes.

What does the graph explain? Figure 1 makes both successes and failures inspectable. In the early repair, the wrong majority has 7 traces but zero excess edges, while the correct candidate has 6 traces and four excess edges. In the later repair, majority is again more frequent (12 versus 8), yet the correct candidate has nine excess edges versus seven, giving a larger normalized score ($9/8$ versus $7/12$). The final pair shows the failure mode: the correct choice at $k-1$ has three excess edges over 12 traces, while at k a wrong candidate has four over 10. Thus a difference of only *one extra excess edge*, amplified by normalization, is a really tricky one.

Together, Figures 1 and 2 support *diversity under*

Changed parameter	Reference	Tested value	Mean gain 5–99	Δ vs. ref. 5–99	Mean gain 61–99	Δ vs. ref. 61–99
Reference configuration			+0.158	0.000	+0.737	+0.000
Community support	5	2	-2.758	-1.326	-4.026	-1.692
		3	-3.568	-2.137	-4.795	-2.462
Precedence support	3	2	-0.289	-0.447	+0.013	-0.724
		5	-0.261	-0.418	+0.045	-0.692
Ordering consistency	0.80	0.60	-0.255	-0.413	-0.167	-0.904
		0.70	+0.032	-0.126	+0.494	-0.244
		0.90	+0.313	+0.155	+1.038	+0.301
Similarity quantile	0.90	0.85	-0.268	-0.412	-0.527	-1.223
		0.95	-0.220	-0.364	-0.227	-0.923
Rolling window	3	1 (static)	+0.121	-0.037	+0.519	-0.218
		5	+0.150	-0.008	+0.673	-0.064
		7	+0.134	-0.024	+0.853	+0.115
		10	+0.079	-0.079	+0.840	+0.103

Table 1: Hyperparameter sensitivity on the training-derived development set. The reference configuration uses community support 5, precedence support 3, ordering consistency 0.80, similarity quantile 0.90, minimum answer frequency 2, and rolling window 5. Each variant changes only the indicated parameter. Gains are accuracy differences relative to majority voting, in percentage points; Δ vs. ref. is the difference in mean gain from the reference. All variants use the same questions and sampling budgets. **Green** refers to improvements over selected hyperparameters.

shared structure: recurrent checkpoints preserve a dominant order, while useful answer families realize those dependencies through multiple compatible routes.

6 Ablation

Table 1 reports one-factor sensitivity on the development set: each variant changes only the indicated hyperparameter, while all others remain fixed to the reference configuration. We report both the full range ($k = 5$ –99) and $k = 61$ –99, since the structural selector provides substantially larger gains over majority voting at higher sampling budgets.

Overall, the fixed configuration is close to a strong operating point: for most hyperparameters, nearby alternatives do not provide consistent improvements. Two exceptions suggest that some parameters are better made budget-dependent. Higher ordering consistency becomes more beneficial at large k , where relations are supported by more traces and stricter agreement can filter occasional reversals without discarding much reliable structure. Longer rolling windows show a similar high-budget advantage. As k grows, the underlying structural signal should change more slowly between consecutive prefixes, while a single added trace can still trigger discrete graph changes by cre-

ating or removing edges or components. Stronger smoothing can suppress this local jitter; at small k , however, each new trace carries substantially more information, so a long window may delay genuine changes in the signal. This suggests increasing both the consistency threshold and smoothing horizon with the sampling budget rather than fixing them globally.

7 Conclusion

Majority voting treats traces as interchangeable votes. We instead expose an answer-local reasoning graph and use its excess connectivity as an interpretable selection signal. On MATH500, this signal becomes informative once enough traces are available and can recover correct minority answers that frequency alone misses. Just as importantly, the same graph reveals when the inductive bias fails. This makes structural self-consistency both a reranking heuristic and an interpretable view of what repeated sampling agrees on.

8 Limitations

Checkpoint graphs inherit errors from segmentation and semantic matching. The answer-local quantile threshold is relative rather than an absolute equivalence probability. Cycle rank measures

redundant connectivity, not logical validity: a coherent misconception can also form cycles, as the drift example illustrates. Results are limited to one generator, one mathematics benchmark, and one fixed sequential sample order; alternative models, domains, temperatures, and trace permutations may change the observed regime. Finally, the rolling score introduces temporal dependence across adjacent prefixes, so per-prefix observations should not be treated as independent statistical tests.

References

- Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *CoRR*, abs/2407.21787.
- Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. [Universal self-consistency for large language model generation](#). *CoRR*, abs/2311.17311.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*, *NeurIPS 2021*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. [Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net.
- Llama Team. 2024. [The llama 3 herd of models](#). *CoRR*, abs/2407.21783.
- Maria Marina, Daniil Moskovskiy, Sergey Pletenev, Mikhail Salnikov, Alexander Panchenko, and Viktor Moskvoretskii. 2026. [Boosting self-consistency with ranking](#). *CoRR*, abs/2606.05054. Also published in the ACL 2026 Student Research Workshop.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. [Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. [Qwen3 embedding: Advancing text embedding and reranking through foundation models](#). *CoRR*, abs/2506.05176.

A Prompts

A.1 Checkpoint-equivalence instruction

The following instruction is supplied to Qwen3-Reranker-0.6B for each ordered query–document pair; both directions are scored and combined by the minimum raw logit.

Judge whether the Document expresses the same mathematical reasoning checkpoint as the Query. Treat paraphrases and algebraically equivalent formulations as matches. The two steps must express the same intermediate claim, mathematical transformation, or result. Reject pairs that merely discuss the same problem or share vocabulary. Differences in numerical value, sign, operator, inequality direction, condition, assumptions, or conclusion mean that the checkpoints are not equivalent.

A.2 MATH500 generation prompt

The prompt follows the generation setup of [Marina et al. \(2026\)](#).

System.

You are a helpful and expert mathematical assistant. Solve the given math problem step-by-step.

User.

Reason step-by-step to solve the math problem. At the end of your solution, strictly follow this format to present your final answer:

Final Answer: $\boxed{\text{answer}}$

Ensure the answer within $\boxed{}$ is concise and contains only the final value or expression.

Problem: {problem}