REASONING CHANGE-POINTS (RCP): TRAINING-FREE FIRST-ERROR LOCALIZATION FROM HIDDEN-STATE DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Hidden-state representations provide a rich substrate for diagnosing multi-step reasoning, yet when reasoning failures leave a reliably localizable boundary in representations remains insufficiently understood. Our key insight is to separate the location of a reasoning error from the representational change that makes that error detectable. We find that boundary strength varies systematically across reasoning regimes: controlled perturbations often induce sharp and reproducible transitions, while naturally occurring failures exhibit subtler structure under passive inspection. Importantly, this structure is not intrinsic to the failure alone: re-evaluating reasoning prefixes under an explicit verification instruction substantially strengthens natural localization, showing that first-error localizability depends not only on the reasoning failure itself but also on the computation used to inspect it. Building on this observation, we introduce **Reasoning Change-Points (RCP)**, a reliability-aware framework that detects local transitions in step-level hidden states, estimates the reliability of candidate error boundaries, and strengthens ambiguous natural evidence through verification conditioning, without training an auxiliary verifier. Across six 4B–8B models spanning multiple model families on GSM8K and MATH-500, controlled failures achieve 0.51–0.90 exact localization, and a prospectively specified Qwen3-32B evaluation reproduces the predicted depth structure. Across four ProcessBench domains, verification conditioning improves exact natural localization by 17.5–22.8 percentage points over passive hidden-state trajectories, indicating confidence calibrated on controlled failures transfers to natural settings, substantially enriching for correctly localized examples. Downstream experiments show that informative boundaries can support targeted regeneration and motivate adaptive reasoning control. Together, these findings characterize a *soft reliability boundary* for hidden-state first-error localization and show how that boundary can be strengthened and selectively exploited when the resulting evidence is reliable.

1 INTRODUCTION

First-error localization is a fundamental requirement for reliable multi-step reasoning. A wrong final answer only reveals that failure occurred somewhere in the reasoning process; diagnosis, verification, and targeted repair require identifying the earliest step after which the trajectory can no longer be trusted. This boundary is especially useful when a system aims to preserve a valid reasoning prefix rather than regenerate the entire solution.

Hidden-state representations have become an increasingly useful source of information for reasoning correctness, hallucination, and reliability diagnosis (Azaria & Mitchell, 2023; Chen et al., 2024; Liang et al., 2026). This naturally raises a more specific question: *when do reasoning failures leave a localizable boundary in the model's hidden-state trajectory?* Our key insight is to distinguish the location of an error from the representational change that makes that error detectable. Correctness-related information can be distributed broadly across a trajectory, whereas first-error localization requires a structured transition that can be associated with a particular reasoning boundary. Understanding when such transitions emerge is therefore central to turning hidden-state evidence into actionable localization.

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.

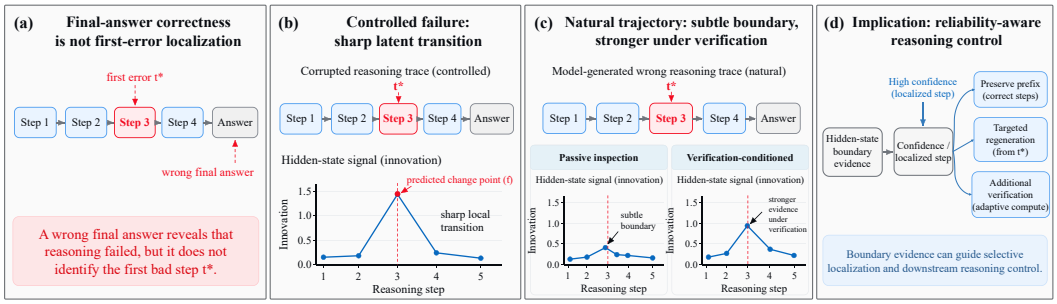

Figure 1: Motivation and key phenomenon. Hidden-state first-error localization operates across a continuum of boundary strength. Controlled failures often produce pronounced local transitions, whereas naturally occurring failures exhibit more varied evidence under passive inspection. Verification conditioning can strengthen this natural boundary evidence. **Reasoning Change-Points** (RCP) exploits the resulting reliability structure to support selective localization and downstream reasoning control.

We find that the strength of these boundaries varies systematically across reasoning regimes. In a controlled setting, a known intermediate step is corrupted while leaving the preceding reasoning prefix unchanged, providing an exact reference location for the introduced failure. Such perturbations often produce sharp and reproducible hidden-state transitions. Naturally occurring reasoning failures , by contrast, exhibit a broader range of transition strengths under passive inspection. We refer to this systematic contrast as the *Controlled–Natural Localization Gap*. Importantly, this gap does not define a binary distinction between localizable and non-localizable errors. Instead, it reveals a continuum in how strongly reasoning failures are expressed as local representational transitions.

Crucially, this boundary strength is not fixed. When the same reasoning prefixes are re-evaluated under an explicit verification instruction, previously subtle natural transitions can become substantially more pronounced. This observation suggests that first-error localizability is shaped jointly by the reasoning failure and the *computational mode used to inspect the trace*. We therefore view hidden-state first-error localization through the lens of a *soft reliability boundary*: different trajectories expose different amounts of localization evidence, and the strength of that evidence can be amplified by changing how the model represents and evaluates the accumulated reasoning.

Building on this observation, we introduce **Reasoning Change-Points (RCP)**, a reliability-aware framework for localizing first reasoning errors from hidden-state trajectories. RCP converts local representational changes into candidate boundaries and evaluates their reliability using complementary trajectory evidence, including transition prominence, cross-layer agreement, and detector agreement. For naturally occurring trajectories, we ask the same frozen model to re-evaluate each reasoning prefix, helping reveal additional boundary structure in the hidden states. This often makes the first-error boundary more visible in the hidden states. RCP thus provides a unified mechanism for identifying, strengthening, and selectively exploiting hidden-state localization evidence, without training an auxiliary verifier.

Our experiments follow this progression from controlled localization to natural failure diagnosis and downstream use. Across six 4B–8B models spanning multiple model families on GSM8K and MATH-500, controlled perturbations produce strong and reproducible localization signals. A prospectively specified evaluation on a held-out Qwen3-32B model further reproduces the predicted depth structure without retuning. Across four ProcessBench reasoning domains, one fixed verification-conditioned protocol consistently strengthens natural first-error localization, improving exact accuracy by 17.5–22.8 percentage points over passive hidden-state trajectories. Confidence estimated only from controlled data further concentrates accurately localized natural cases without natural-label tuning. Finally, targeted regeneration experiments demonstrate that informative boundaries can support downstream correction, suggesting a path toward adaptive reasoning systems that allocate verification and regeneration according to internal boundary evidence.

Our contributions are threefold:

- **A reliability-centered view of hidden-state localization.** We show that reasoning errors differ in how clearly they appear as boundaries in hidden-state trajectories. Controlled errors usually produce stronger boundaries than naturally occurring errors, and these boundaries can become clearer when the model is explicitly asked to verify the reasoning.
- **Reasoning Change-Points (RCP)**. We introduce RCP, a framework that detects candidate error boundaries from step-level hidden-state transitions, and estimates how reliable each boundary is. For natural errors, RCP further strengthens the boundary signal by re-evaluating each reasoning prefix with the same frozen model, without training a separate verifier.
- **Evidence from localization to downstream control.** Through controlled multi-backbone experiments, a prospective 32B test, four ProcessBench domains, controlled-to-natural confidence transfer, and targeted regeneration, we show that hidden-state boundary evidence can be reproducibly detected, computationally strengthened, and selectively exploited for downstream reasoning control.

2 Problem Formulation

We study *first-error localization*. For a multi-step reasoning trace $\tau = (s_1, \ldots, s_K, a)$, the target $t^\star$ is the earliest step whose reasoning is invalid under the task semantics. For a frozen language model, the representation of step t at layer ℓ and inspection mode m is $h_t^{(\ell,m)} \in \mathbb{R}^H$.

We consider two representations of the same reasoning process. A *generation-conditioned* trajectory comes from the model's evolving solution. A *verification-conditioned* trajectory re-evaluates accumulated prefixes under an instruction to assess whether the shown reasoning contains an error. A change-point rule maps either trajectory to a candidate location $\hat{t}$, allowing us to study both where a boundary appears and how its strength changes with the inspection computation.

For each candidate boundary, RCP also computes reliability evidence $C(\tau)$. An acceptance decision $A(\tau) \in \{0, 1\}$ determines whether the localized boundary enters the selected operating region. The corresponding operating point is characterized jointly by coverage, $\Pr[A(\tau) = 1]$, and localization accuracy among accepted cases. In this way, localization is treated as a structured decision over both *where* a boundary occurs and *how strongly* the trajectory supports that boundary.

We evaluate this formulation in two complementary regimes. In the *controlled* regime, an otherwise correct trace is corrupted at a known eligible reasoning step while the preceding prefix is preserved exactly, yielding an unambiguous reference point for studying the emergence of hidden-state transitions. In the *natural* regime, no error is injected and first-error labels come from ProcessBench. Together, the two regimes expose how localization structure changes from externally anchored failures to naturally generated reasoning.

3 Reasoning Change-Points

RCP treats first-error localization as a problem of detecting reliable changes in the hidden-state trajectory of a reasoning process. It first identifies candidate boundaries from local representational changes, then evaluates whether the surrounding trajectory provides consistent evidence that the change marks a true error boundary. Controlled perturbations provide exact reference points for studying and calibrating this structure, while verification conditioning offers a way to amplify natural boundary evidence using the same frozen model.

3.1 Controlled reference construction

We begin with a correct reasoning trace divided into steps and choose an intermediate step $t^\star$ to perturb (never the first step). We consider three types of corruption: arithmetic corruption changes the result of a local computation, operator swap replaces one arithmetic operator with another, and quantity swap replaces a value with another quantity appearing in the same problem. The corruption is introduced exactly at $t^\star$ while all preceding tokens remain unchanged; generation then continues from the corrupted prefix.

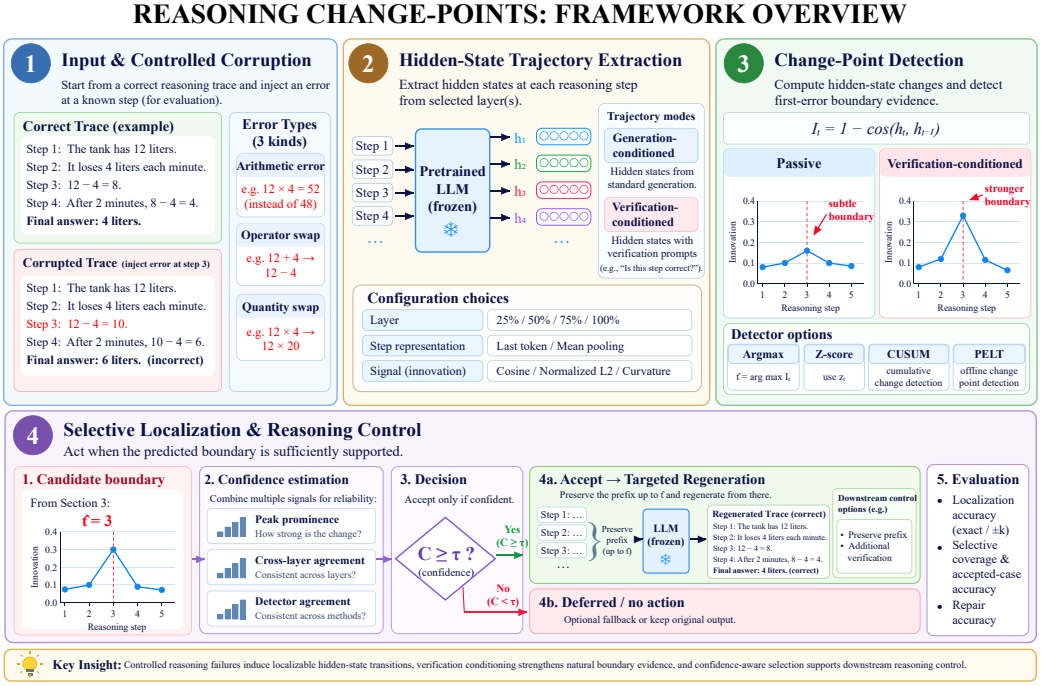

Figure 2: Overview of **Reasoning Change-Points (RCP)**. Controlled perturbations provide known error locations for studying hidden-state transitions. RCP extracts generation- and verification-conditioned trajectories, converts local representational changes into candidate boundaries, and evaluates their reliability using complementary trajectory evidence. Verification conditioning can strengthen natural boundary evidence, while sufficiently supported boundaries can guide targeted regeneration and other downstream reasoning-control decisions.

This construction gives us an exact reference location for the first introduced error. Before $t^\star$, the clean and corrupted traces are identical, so any change that appears at the intervention point cannot be explained by earlier differences in the text. The controlled regime therefore lets us directly study whether a known reasoning error produces a detectable change in the model's hidden-state trajectory.

3.2 Hidden-State Trajectory and Candidate Boundary

For each reasoning step, we extract a hidden-state representation using either the final token of the step or mean pooling over its tokens. To compare architectures with different numbers of transformer blocks, layers are indexed by relative depth $\{25, 50, 75, 100\}\%$. RCP measures local trajectory change with cosine innovation, $I_t = 1 - \cos(h_t, h_{t-1})$, and considers normalized ℓ_2 displacement and trajectory curvature as alternatives.

Because reasoning traces are typically short, we compare several simple change-point rules, including direct peak rules with z-score normalization, CUSUM, and PELT (Killick et al., 2012). These detectors do not train a separate verifier; they convert the observed change sequence into a candidate boundary $\hat{t}$. This design keeps the localization signal tied directly to the geometry of the frozen model's trajectory.

3.3 Reliability Boundary and Selective Localization

Not every large hidden-state change corresponds to the first reasoning error. RCP therefore estimates how much evidence supports each candidate boundary. Peak prominence P measures how strongly the selected transition stands out from the remainder of the trace, cross-layer agreement L measures whether different relative depths support a similar location, and detector agreement D measures

whether different change-point rules select similar locations. We use the fixed combination $C = 0.4P + 0.3L + 0.3D$ for the controlled selective analysis.

We use the confidence score to decide which predicted boundaries are reliable enough to accept. To choose this threshold, We deterministically split controlled GSM8K trajectories into calibration and evaluation halves. On the calibration half, we choose the threshold that keeps as many examples as possible while ensuring that the 95% bootstrap lower bound on exact localization accuracy is at least 0.80. We then apply this threshold unchanged to the held-out controlled examples. Finally, we transfer the same confidence rule to natural reasoning errors without tuning on ProcessBench labels, allowing us to test whether confidence learned from controlled errors also identifies reliable boundaries in natural failures.

3.4 VERIFICATION CONDITIONING

Natural reasoning errors often produce weaker hidden-state boundaries than controlled perturbations. To strengthen these signals, we ask the same frozen model to re-evaluate the reasoning it has produced.

For each prefix $(s_1, \ldots, s_t)$, we provide the original problem together with the reasoning generated up to step t and ask the model whether the shown reasoning contains a mathematical or logical error. We then extract the final-layer hidden state at the end of this verification prompt. Repeating this procedure for every prefix gives a verification-conditioned hidden-state trajectory, $h_t^{(\ell,\mathrm{ver})}$

The key idea is that the model now represents the same reasoning prefix while explicitly evaluating its correctness. This can make a previously weak error boundary more visible in hidden-state space, while leaving the model parameters unchanged. No ProcessBench first-error labels are used to train or tune this representation. We use the same verification prompt, final-layer representation, and hidden-state change rule across GSM8K, MATH, OmniMath, and OlympiadBench, without domain-specific retuning.

For $t > 1$, the verification change score compares prefixes ending at steps $t - 1$ and t, aligning the transition with the newly appended reasoning step.

3.5 TARGETED INTERVENTION

Once RCP identifies a sufficiently reliable boundary $\hat{t}$, we can use that location to guide correction. We truncate the reasoning immediately before $\hat{t}$, keep the preceding prefix unchanged, and ask the same frozen model to regenerate the remaining reasoning and final answer.

This intervention is intentionally minimal: it tests whether localized internal structure can serve as a control signal for downstream reasoning. More broadly, the same reliability estimate could be used to decide when a model should spend additional computation on verification or regeneration, allowing extra inference effort to be focused on trajectories that show strong evidence of a localized failure.

4 EXPERIMENTS

The experiments are organized around the emergence, strengthening, and use of hidden-state boundary evidence. We first test whether known failures create reproducible local transitions across model families. We then evaluate whether a structural depth pattern survives a prospectively frozen 32B experiment. Next, we test whether verification conditioning strengthens natural first-error localization across four ProcessBench domains. Finally, confidence transfer and targeted regeneration evaluate whether the resulting boundary evidence can support selective downstream decisions.

4.1 EXPERIMENTAL SETUP

Models and controlled tasks. We evaluate Qwen3-4B/8B, Llama-3-8B-Instruct, Mistral-7B-Instruct, Phi-4-mini-instruct, and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) on GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021). Models are frozen and run in bf16 with greedy decoding. A held-out Qwen3-32B run is reserved for the prospective depth test.

Table 1: Controlled first-error localization across six 4B–8B models and two benchmarks. **Cfg.** abbreviates depth/signal/detector for the strongest configuration; **Best** E and ± 1 report its exact and one-step localization, while **Fixed** E uses the common 50%-depth/last-token/cosine/argmax readout. Full configuration landscapes appear in Appendices C and T.1.

Benchmark	Model	Cfg.	Best E / ± 1	Fixed E
GSM8K	Qwen3-4B	100/cos/arg	.80 / .93	.66
	Qwen3-8B	100/cos/zsc	.77 / .88	.62
	Llama-3-8B	100/cos/arg	.71 / .87	.34
	Mistral-7B	25/norm/arg	.87 / .97	.72
	Phi-4-mini	100/norm/cus	.62 / .87	.40
	DS-R1-7B	100/norm/cus	.80 / .96	.54
MATH-500	Qwen3-4B	100/cos/zsc	.62 / .84	.48
	Qwen3-8B	100/cos/arg	.77 / .88	.55
	Llama-3-8B	100/cos/zsc	.55 / .81	.30
	Mistral-7B	100/norm/cus	.90 / .96	.70
	Phi-4-mini	100/cos/cus	.51 / .75	.36
	DS-R1-7B	100/cos/cus	.77 / .89	.52

Natural evaluation. For natural localization, we fix Qwen3-8B as the primary inspection backbone and evaluate ProcessBench (Zheng et al., 2025) across GSM8K, MATH, OmniMath, and Olympiad-Bench. ProcessBench provides human-annotated first-error positions for model-generated traces. One common verification-conditioned protocol is applied across all four domains: the prompt, final-layer representation, and hidden-state change rule are not tuned separately for any domain. A separate Qwen3-32B GSM8K stress test repeats the same verification protocol to probe cross-scale persistence within the Qwen family.

Configurations and metrics. A controlled geometric configuration consists of relative depth, step representation, innovation signal, and change-point rule. Main controlled results report the strongest configuration for each model–benchmark pair, while the appendix reports a fixed configuration and complete grids. Natural verification-conditioned results use the frozen final-layer hidden-state change rule. We report exact accuracy and ± 1 accuracy; selective localization additionally reports coverage and accepted-case accuracy. Fixed-configuration and complete controlled configuration landscapes are reported in Appendices C and T.1; additional verification-conditioned natural analyses, confidence transfer, and cross-scale results are reported in Appendix G.

4.2 Controlled failures reveal reproducible change points

Table 1 shows a strong controlled effect across architectures. Exact localization ranges from 0.51 to 0.90 over the twelve model–benchmark cells, and ± 1 accuracy reaches 0.97. The repeated pattern across models with different architectures, depths, and reasoning styles indicates that a known reasoning perturbation can induce a recoverable local transition in hidden-state space.

Table 1 also reports a common fixed readout, showing that substantial localization is already visible without per-model configuration selection. Held-out configuration selection on disjoint halves further preserves strong localization across all twelve model–benchmark cells. In a separate frozen three-seed Qwen3-8B/GSM8K campaign, exact localization is 0.828/0.809/0.796 across seeds, with the final-depth readout strongest in all three runs (Appendix T.5). Together, these results show a reproducible change-point structure across architectures, benchmarks, configurations, and random seeds.

4.3 The transition structure generalizes prospectively

To test whether the observed depth structure supports prospective prediction, we froze two directional predictions from the six smaller models before executing a Qwen3-32B run: localization should strengthen toward the final depth, and mean fusion should not outperform the strongest final-depth signal. Under the fixed configuration, exact localization rises monotonically from 0.482 at 25% depth to 0.570, 0.628, and 0.725 at 50/75/100%, while mean fusion reaches 0.650. Both directional predictions are satisfied.

This prospective result reveals a coarse depth structure in the localization signal that survives a substantial scale increase without retuning. It complements the multi-model controlled study by

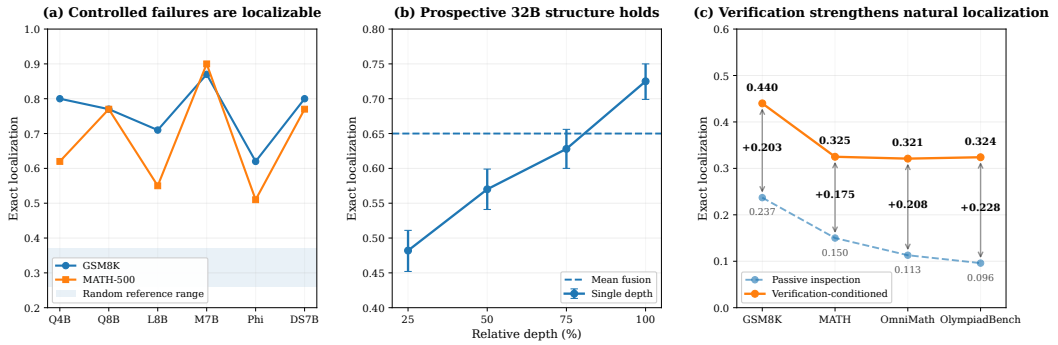

Figure 3: Evidence across controlled, prospective, and natural reasoning regimes. (a) Controlled first-error localization is strong across six 4B–8B models on GSM8K and MATH-500; the shaded band spans the corresponding per-trajectory random-reference range. (b) On the prospectively specified Qwen3-32B evaluation, exact localization increases with relative depth and the final-layer signal exceeds mean fusion, matching both pre-specified directional predictions; error bars denote 95% bootstrap confidence intervals. (c) Across four ProcessBench domains, verification conditioning consistently strengthens natural first-error localization, improving exact accuracy by 17.5–22.8 percentage points over passive hidden-state trajectories.

Table 2: Verification-conditioned natural first-error localization and transfer diagnostics. Top: one fixed Qwen3-8B inspection protocol across four ProcessBench domains, with gain over passive inspection and selective exact accuracy/coverage after transferring the controlled threshold. Middle: the same protocol on Qwen3-32B GSM8K. Bottom: Qwen3-8B localization decomposed by GSM8K trace generator.

Inspector / subset	Domain	n	Verif. E	±1	Gain	Sel. E/Cov.
Qwen3-8B	GSM8K	207	.440	.744	+.203	.512/.396
	MATH	594	.325	.588	+.175	.391/.332
	OmniMath	759	.321	.547	+.208	.346/.141
	OlympiadBench	661	.324	.516	+.228	.429/.138
Qwen3-32B	GSM8K	207	.575	.783	—	.692/.440
Qwen2 traces	GSM8K	77	.416	.753	+.208	.462/.506
Qwen2.5 traces	GSM8K	58	.397	.690	+.172	.500/.172
Llama traces	GSM8K	45	.511	.778	+.200	.545/.489
Meta-Llama traces	GSM8K	27	.481	.778	+.259	.636/.407

showing that the geometry is not only detectable retrospectively but can also support directional predictions about where localization evidence becomes strongest.

4.4 Verification conditioning amplifies natural boundary evidence

We next ask whether boundary strength can be changed by the computation used to inspect a naturally generated reasoning trace. With Qwen3-8B fixed as the inspection backbone, we re-evaluate every reasoning prefix under one explicit verification instruction and extract the final-layer hidden representation at the verification boundary. The same prompt, representation, and change rule are used across all four ProcessBench domains without domain-specific tuning.

Verification conditioning consistently strengthens exact first-error localization. Exact accuracy reaches 0.440 on GSM8K, 0.325 on MATH, 0.321 on OmniMath, and 0.324 on OlympiadBench, corresponding to absolute improvements of 17.5–22.8 percentage points over passive hidden-state trajectories. The consistency of this gain across four reasoning domains supports a central conclusion of RCP: first-error localizability is not fixed by the original generation trajectory, but can be amplified by changing the model's inspection computation.

The controlled confidence score also transfers to natural reasoning without ProcessBench calibration, concentrating more accurate localizations on all four domains. The same verification protocol scales to Qwen3-32B on ProcessBench GSM8K, reaching 0.575 exact and 0.783 within-one-step

Table 3: From localization to intervention. Top: controlled GSM8K targeted regeneration; ΔOrig is the gain over the original corrupted trace and ΔRand is the accuracy difference from random-cut regeneration. Middle: held-out selective operating points. Bottom: same-model natural prefix-preserving repair with Qwen3-8B.

Controlled targeted regeneration					
Model	Original	RCP cut	ΔOrig	ΔRand	Oracle cut
Qwen3-8B	.560	.727	+.167	+.214	.853
Mistral-7B	.467	.707	+.240	+.334	.740
Qwen3-4B	.540	.587	+.047	+.134	.707
Held-out selective localization					
Model	τ	Coverage	Accepted E	95% LB	–
Qwen3-8B	.775	.864	.836	.806	–
Mistral-7B	.850	.868	.851	.817	–
Same-model natural repair (276 incorrect Qwen3-8B GSM8K traces)					
Cut policy	Accuracy	Gain	95% CI	Wins/Losses	–
Passive hidden-state	.326	ref.	—	—	—
Verification-conditioned	.442	+.116	[.062,.170]	46/14	–

localization; the transferred threshold selects 44.0% of traces at 0.692 exact accuracy. The generator-family decomposition in Table 2 further shows that the verification-conditioned signal appears across Qwen- and Llama-generated traces, while Appendix G provides the full protocol and observability analysis.

4.5 Localized boundaries support targeted regeneration

We finally test whether localized boundaries can serve as useful downstream control signals. On controlled GSM8K failures, RCP truncation improves Qwen3-8B answer accuracy from 0.560 to 0.727 and Mistral-7B from 0.467 to 0.707, recovering substantial oracle-cut headroom. Qwen3-4B also improves from 0.540 to 0.587. These results show that when a boundary is informative, preserving the preceding prefix and regenerating only the suffix can improve final-answer correctness.

The same controlled study also yields high-coverage selective operating points: Qwen3-8B and Mistral-7B retain 86.4% and 86.8% coverage while reaching 0.836 and 0.851 accepted-case exact localization, respectively. A separate same-model natural repair study on 276 incorrect Qwen3-8B GSM8K generations shows the same direction of improvement downstream: verification-conditioned cuts raise prefix-preserving repair accuracy from 0.326 to 0.442 (+11.6 percentage points). Protocol details and paired-bootstrap analysis are reported in Appendix Q.2. Together, these results connect localized hidden-state boundaries to targeted regeneration and reliability-aware reasoning control.

5 Analysis: What Shapes First-Error Localizability?

The experiments reveal that first-error localizability is a structured property of the interaction between failure type, representation dynamics, and inspection computation.

Controlled perturbations expose a reproducible localization regime. The bit-identical pre-intervention prefix makes the controlled setting a clean reference for studying the emergence of a local transition. Across model families, the intervention produces recoverable boundary structure, and the prospective 32B experiment further reveals a predictable depth organization. These results establish that first-error boundaries are not merely artifacts of one model or one detector configuration.

Inspection computation changes boundary strength. Verification conditioning substantially strengthens natural first-error localization across four ProcessBench domains under one fixed protocol. The same reasoning prefix can therefore occupy different localization regimes depending on whether it is represented during ordinary generation or explicit verification. This suggests that hidden-state localizability is not solely a static property of the trace: it is partly an emergent property of the computation used to interrogate that trace.

Reliability converts localization into a control signal. Confidence transferred from controlled references concentrates more accurate natural localizations, while high-confidence controlled boundaries support targeted regeneration. This suggests a broader role for RCP beyond error diagnosis. Hidden-state boundary evidence can potentially determine when a system should preserve a prefix, invoke additional verification, allocate more inference compute, or regenerate a suffix. Such adaptive policies turn local internal geometry into a mechanism for reasoning-time control.

6 Related Work

Process-level verification and self-correction. Intermediate reasoning is evaluated through self-correction (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2023), process reward models (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024; Zhang et al., 2025; Zhao et al., 2025), and step-level benchmarks (Zheng et al., 2025; He et al., 2025; Song et al., 2025; Xia et al., 2025). Complementary neuro-symbolic approaches combine neural reasoning with symbolic validation and feedback-guided refinement to improve reasoning reliability (Tian et al., 2026). These lines of work establish the value of intermediate verification and structured correction. RCP instead asks whether the model's own hidden-state trajectory exposes a local boundary at the first reasoning error, and how that boundary can be strengthened and used for selective downstream action.

Internal-state signals. Hidden activations support truthfulness, hallucination, correctness, and inference-time diagnostics (Azaria & Mitchell, 2023; Burns et al., 2023; Li et al., 2023; Chen et al., 2024; Chuang et al., 2024), including reasoning-level reliability analyses such as CLUE and HARP (Liang et al., 2026; Hu et al., 2026). Concurrent transport-geometry work studies step-level hallucination detection with learned classifiers (Alvarez & Baheri, 2026). RCP instead treats the reasoning trace itself as a short hidden-state trajectory and studies the emergence, strengthening, and reliability of local change points without training an auxiliary probe.

Change-point detection and controlled perturbations. Classical change-point methods use local, cumulative, or segmentation statistics (PAGE, 1954; Adams & MacKay, 2007; Killick et al., 2012; Truong et al., 2020), with related sequential analyses of generation outcomes (Bigelow et al., 2024). Because reasoning traces are short, we compare simple peak rules with CUSUM and PELT. Our intervention protocol is also related to controlled reasoning-robustness studies (Mirzadeh et al., 2025; Lanham et al., 2023; Turpin et al., 2023), but uses the known intervention point as an exact anchor for studying hidden-state localization.

7 Limitations and Conclusion

The primary natural study uses Qwen3-8B, with Qwen3-32B establishing scale persistence only within the Qwen family. ProcessBench traces are not always generated by the inspected model, and verification conditioning introduces additional inference cost. The current experiments also focus on mathematical and structured reasoning traces, so it remains unclear how strongly the same boundary structure extends to longer-form reasoning, open-domain tasks, or settings in which the first error is semantically ambiguous. Future work should therefore test same-checkpoint human-annotated failures, broader model families, more diverse reasoning domains, and more efficient verification schedules. It would also be useful to study whether boundary reliability can be estimated online during generation rather than only after a complete trace is available.

RCP characterizes first-error localization as a reliability-structured hidden-state phenomenon rather than a universally sharp signal. Controlled failures yield reproducible change points, verification conditioning strengthens natural boundaries, and controlled-to-natural confidence transfer identifies operating regions in which localized boundaries can support targeted regeneration. Taken together, these results suggest that hidden-state geometry is most useful when treated as graded evidence rather than as an unconditional error detector. This perspective supports a selective use of internal signals: strong boundaries can trigger focused verification or regeneration, whereas weak boundaries can remain deferred rather than forcing an unreliable localization decision. These results position hidden-state change points as a practical signal for selective verification, prefix preservation, suffix regeneration, and adaptive allocation of reasoning-time computation.

AI USE STATEMENT

Generative AI tools assisted with language editing and LaTeX formatting. The authors reviewed all scientific claims, experimental results, citations, and AI-assisted text and take responsibility for the final content of the submission.

REPRODUCIBILITY STATEMENT

The main text specifies controlled corruption, hidden-state extraction, innovation signals, detector families, reliability-aware selection, verification conditioning, evaluation settings, and targeted regeneration. The appendix provides fixed and complete configuration landscapes, held-out configuration selection, statistical procedures, verification-conditioned natural analyses, seed robustness checks, prompt and protocol details, observability analyses, and machine-readable artifact indices.

REFERENCES

Ryan Prescott Adams and David J. C. MacKay. Bayesian online changepoint detection, 2007. URL https://arxiv.org/abs/0710.3742.

Tyler Alvarez and Ali Baheri. Where does reasoning break? step-level hallucination detection via hidden-state transport geometry, 2026. URL https://arxiv.org/abs/2605.13772.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Eric Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text generation, 2024. URL https://arxiv.org/abs/2412.07961.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Zj12nzlQbz.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

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.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *Second Agent Learning in Open-Endedness Workshop*, 2023. URL https://openreview.net/forum?id=WSrRF5Wy6v.

Daya Guo, Dejian Yang, Haowei Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Z.Y. Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, and Bo Zheng. Can large language models detect errors in long chain-of-thought reasoning? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18468–18489, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.905. URL https://aclanthology.org/2025.acl-long.905/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Junjie Hu, Gang Tu, Cheng Shengyu, JinXin Li, Jinting Wang, Rui Chen, Zhilong Zhou, and Dongbo Shan. Harp: Hallucination detection via reasoning subspace projection. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 17783–17802, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1e58b1bf9f218fcd19e4539e982752a5-Paper-Conference.pdf.

Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/81b8390039b7302c909cb769f8b6cd93-Paper-Conference.pdf.

Zhenwen Liang, Ruosen Li, Yujun Zhou, Linfeng Song, Dian Yu, Xinya Du, Haitao Mi, and Dong Yu. Your reasoning model is secretly a reward model - optimization-free verification from experience. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17358–17372, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.788. URL https://aclanthology.org/2026.acl-long.788/.

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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

E. S. PAGE. Continuous inspection schemes. *Biometrika*, 41(1-2):100–115, 06 1954. ISSN 0006-3444. doi: 10.1093/biomet/41.1-2.100. URL https://doi.org/10.1093/biomet/41.1-2.100.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained and challenging benchmark for process-level reward models, 2025. URL https://arxiv.org/abs/2501.03124.

Jinlong Tian, Jiang Yu, Kewei Cheng, Fengxiang Cheng, Yue He, Yunfei Wang, Haotian Wang, Haoxuan Li, Wenjing Yang, and Shixuan Liu. LogicSAGE: Neuro-symbolic reasoning with socratic-guided enhancement. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=GxSB5ZYfM4.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020. ISSN 0165-1684. doi: 10.1016/j.sigpro.2019.107299. URL https://www.sciencedirect.com/science/article/pii/S0165168419303494.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

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

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. 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)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27723–27730, 2025.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning, 2025. URL https://arxiv.org/abs/2501.07301.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. Genprm: Scaling test-time compute of process reward models via generative reasoning, 2025. URL https://arxiv.org/abs/2504.00891.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning, 2025. URL https://arxiv.org/abs/2412.06559.

A Appendix Overview

The supplementary material is organized around the same progression as the main paper: *boundary emergence*, *structural generalization*, *verification-conditioned strengthening*, *reliability transfer*, and *downstream control*. We retain detailed implementation, configuration, statistical, and reproducibility information while using the main appendix tables to emphasize the operating regimes that support the paper's claims. Exhaustive machine-readable outputs remain bound to the released artifacts through the hashes listed at the end of the appendix.

B Implementation Details

B.1 Generation and Decoding Settings

All models are evaluated in bf16 with greedy decoding (temperature 0). Each model is prompted with the instruction "Solve the following math problem step by step. End with 'Final Answer: ¡answer¿'.", followed by the question. Reasoning traces are parsed into numbered steps by splitting on the pattern `Step k:`; the final answer is extracted from the `Final Answer:` line. The distilled-R1 model's `<think>` block is excluded from step parsing, consistent with the main text. Only trajectories that parse cleanly and contain a recoverable final answer are retained.

B.2 Relative Depth to Layer Index Mapping

We read hidden states at relative depths $\{25\%, 50\%, 75\%, 100\%\}$ of each model's depth. The absolute layer indices follow the implementation rule $\mathrm{round}(dL)$ clipped to $[1, L]$: Qwen3-4B (layers 9/18/27/36 of 36), Qwen3-8B (9/18/27/36 of 36), Llama-3-8B (8/16/24/32 of 32), Mistral-7B (8/16/24/32 of 32), Phi-4-mini (10/20/30/40 of 40), DeepSeek-R1-Distill-Qwen-7B (7/14/21/28 of 28), and Qwen3-32B (16/32/48/64 of 64). All depths are extracted in a single forward pass with hidden-state outputs enabled.

Table 4: Model inventory and relative-depth layer mapping used throughout the experiments.

Model	Layers L	25/50/75/100% indices	Experimental role
Qwen3-4B	36	9/18/27/36	Controlled GSM8K/MATH-500
Qwen3-8B	36	9/18/27/36	Controlled + primary natural inspector
Llama-3-8B	32	8/16/24/32	Controlled GSM8K/MATH-500
Mistral-7B	32	8/16/24/32	Controlled GSM8K/MATH-500
Phi-4-mini	40	10/20/30/40	Controlled GSM8K/MATH-500
DS-R1-7B	28	7/14/21/28	Controlled GSM8K/MATH-500
Qwen3-32B	64	16/32/48/64	Prospective + natural cross-scale

B.3 Controlled Corruption Protocol

For each valid clean trace, candidate injection steps are reasoning steps 2 through K; step 1 is excluded because no preceding inter-step change exists. Candidates are ordered by distance to the trajectory midpoint, with earlier steps breaking ties, and the corruption operators are tried in a fixed priority order. We apply one of three corruption operators: (i) *arithmetic* corruption perturbs the right-hand side of an equation $a \circ b = c$; (ii) *operator swap* replaces $+ \leftrightarrow -$ or $\times \leftrightarrow \div$; (iii) *quantity swap* replaces a number with another quantity occurring in the same problem. The edited step is spliced at the token boundary immediately before the heading `Step` $t^\star$`:`, so the clean and corrupted token prefixes are bit-identical before $t^\star$. Generation continues from the corrupted step with the same model and decoding settings.

B.4 Repair Procedure Details

For repair we truncate the corrupted trace immediately before the predicted step $\hat{t}$, keep the prefix up to $\hat{t} - 1$, append the heading `Step` $\hat{t}$`:`, and regenerate the suffix with up to 512 new tokens using the same frozen model and greedy decoding. The **self-reflection** reference uses a multi-turn prompt: the

Table 5: Controlled corruption operators. Every operator changes exactly one eligible reasoning step while preserving the preceding token prefix.

Operator	Local edit
Arithmetic perturbation	Change the computed right-hand-side value of a local arithmetic expression.
Operator swap	Replace $+ \leftrightarrow -$ or $\times \leftrightarrow \div$ at the selected step.
Quantity swap	Replace a numeric quantity with another quantity appearing in the same problem.

original instruction and the corrupted trace are followed by the user message "Review your reasoning above. If any step is wrong, correct it and continue. End with 'Final Answer: ¡answer¿'.", and the model regenerates with up to 512 new tokens.

Table 6: Repair and regeneration settings used in the downstream experiments.

Setting	Controlled repair	Same-model natural repair
Checkpoint	Same frozen generator	Qwen3-8B throughout
Prefix handling	Preserve through $\hat{t} - 1$	Preserve through predicted cut
Decoding	Greedy	Greedy
Maximum new tokens	512	384

C Complete Fixed-Configuration Results

Table 7 reports every model under the single fixed configuration (50% depth, last-token representation, cosine innovation, argmax detection) so that the main-text phenomenon can be evaluated without any per-model configuration search.

Table 7: Fixed-configuration localization (50% depth, last token, cosine innovation, argmax).

	GSM8K		MATH-500	
Model	E	± 1	E	± 1
Qwen3-4B	.66	.90	.48	.76
Qwen3-8B	.62	.85	.55	.84
Llama-3-8B	.34	.72	.30	.66
Mistral-7B	.72	.93	.70	.88
Phi-4-mini	.40	.77	.36	.68
DS-R1-7B	.54	.84	.52	.79
Random	.34–.37	.77–.83	.26–.34	.64–.77

D Configuration Grid Ablations

The controlled sweep crosses four relative depths, two step representations (last-token and mean-pooling), three geometric signals (cosine innovation, normalized ℓ_2 displacement, and trajectory curvature), and four short-series detectors (argmax, z-score, CUSUM, and PELT). Table 9 summarizes the strongest configuration for each model–benchmark pair; the complete last-token and mean-pooling landscapes are reported in Appendix T.1.

The strongest readout is concentrated at the final depth in 11 of 12 model–benchmark pairs, with Mistral-7B/GSM8K as the shallow-depth exception. Cosine and normalized-ℓ_2 transitions account for all selected configurations, while argmax, z-score, and CUSUM provide complementary short-series readouts. This pattern motivates the depth analysis below and the prospective Qwen3-32B test.

Table 8: Controlled configuration space. The Cartesian product contains 96 geometric configurations per model–benchmark pair before fusion variants.

Dimension	Count	Options
Relative depth	4	25%, 50%, 75%, 100%
Step representation	2	Last token, mean pooling
Geometric signal	3	Cosine, normalized ℓ_2, curvature
Detector	4	Argmax, z-score, CUSUM, PELT
Total	96	per model–benchmark pair

Table 9: Best controlled configuration and localization accuracy for each model–benchmark pair. "Cfg." abbreviates depth/signal/detector.

Model	Benchmark	Cfg.	Exact	± 1
Qwen3-4B	GSM8K	100/cos/arg	.799	.932
Qwen3-4B	MATH-500	100/cos/zsc	.624	.844
Qwen3-8B	GSM8K	100/cos/zsc	.770	.883
Qwen3-8B	MATH-500	100/cos/arg	.767	.876
Llama-3-8B	GSM8K	100/cos/arg	.710	.873
Llama-3-8B	MATH-500	100/cos/zsc	.545	.805
Mistral-7B	GSM8K	25/norm/arg	.875	.970
Mistral-7B	MATH-500	100/norm/cus	.900	.960
Phi-4-mini	GSM8K	100/norm/cus	.616	.866
Phi-4-mini	MATH-500	100/cos/cus	.507	.750
DS-R1-7B	GSM8K	100/norm/cus	.802	.956
DS-R1-7B	MATH-500	100/cos/cus	.774	.887

E Depth–Fusion Interaction

Layer depth is one of the clearest structural axes in the controlled study. Table 10 reports the canonical GSM8K cosine/argmax depth profile. Five of six models achieve their strongest canonical readout at the final depth, while Mistral-7B maintains a strong signal throughout the stack.

Table 10: GSM8K exact localization across relative depths under the canonical last-token cosine/argmax view.

Model	25%	50%	75%	100%
Qwen3-4B	.466	.466	.554	.799
Qwen3-8B	.555	.623	.663	.770
Llama-3-8B	.364	.368	.415	.710
Mistral-7B	.700	.716	.677	.792
Phi-4-mini	.377	.331	.529	.487
DS-R1-7B	.525	.497	.585	.711

We also compare two ways to aggregate per-depth innovation series before detection. Mean aggregation is competitive with or stronger than max aggregation in most tested cells, with the largest mean–max difference equal to 0.065. We therefore use mean fusion as the default multi-depth summary while treating the single-depth profile as the primary structural diagnostic.

F Prospective Test on a Held-Out 32B Model

To test whether the observed depth structure generalizes prospectively, we froze two directional rules on the six smaller models before executing Qwen3-32B: (P1) a monotone depth profile predicts that the final depth will be strongest; and (P2) a shallow-profile rule predicts whether multi-depth fusion will improve on the strongest single depth. Table 12 reports the held-out result.

Table 11: Fusion-mean vs. fusion-max exact accuracy on GSM8K with CUSUM.

	Cosine innovation		Normalized ℓ_2	
Model	Mean	Max	Mean	Max
Qwen3-4B	.515	.502	.490	.458
Qwen3-8B	.614	.554	.617	.552
Llama-3-8B	.467	.497	.466	.492
Mistral-7B	.817	.792	.843	.813
Phi-4-mini	.414	.370	.386	.357

Table 12: Qwen3-32B GSM8K prospective depth test ($n = 1119$ valid pairs).

Readout	Exact	95% bootstrap CI
25% depth	.482	[.452,.511]
50% depth	.570	[.541,.599]
75% depth	.628	[.600,.656]
100% depth	.725	[.699,.750]
Mean fusion	.650	[.624,.680]

Exact localization increases monotonically from 0.482 to 0.725 with depth, while mean fusion reaches 0.650. Both pre-specified directional predictions match the observed outcomes. The frozen rule file, evaluation script, and result artifacts are hashed and archived with the submission.

G Verification-Conditioned Natural Localization

The natural study evaluates whether first-error boundary evidence can be strengthened when the same reasoning prefix is represented under an explicit verification computation. Qwen3-8B is fixed as the primary inspection backbone and the same native-chat verification instruction, final-layer hidden representation, and consecutive-prefix cosine change rule are used across GSM8K, MATH, OmniMath, and OlympiadBench. No ProcessBench first-error label is used to train an auxiliary verifier or to tune the representation separately for any domain.

Table 13: Frozen natural-localization protocol shared across all four ProcessBench domains.

Protocol element	Setting
Inspection backbone	Qwen3-8B
Prompt mode	Native-chat verification instruction
Representation	Final-layer hidden state at verification boundary
Change score	Consecutive-prefix cosine innovation
Domain-specific tuning	None
First-error supervision for representation	None

G.1 Four-Domain Verification Protocol

For each annotated reasoning trace, we form one accumulated prefix after every reasoning step. Each prefix is re-evaluated by the same frozen model under an instruction asking whether the reasoning shown so far contains a mathematical or logical error. Let v_t denote the final-layer hidden representation at the verification boundary for the prefix ending at step t. We score the transition into step t by

$$J_t = 1 - \cos(v_t, v_{t-1}), \qquad t > 1,$$

and predict the largest eligible transition. The representation, layer, prompt family, and change rule are held fixed across all four ProcessBench domains.

Table 14 reports the verification-conditioned results together with the absolute improvement over the corresponding passive trajectory. We emphasize the cross-domain consistency of the improvement rather than any single benchmark value.

Table 14: Verification-conditioned natural first-error localization with Qwen3-8B fixed as the inspection backbone. Gain is the absolute improvement over the corresponding passive hidden-state trajectory.

Domain	Verif. exact	Verif. ± 1	Gain vs. passive
GSM8K	.440	.744	+.203
MATH	.325	.588	+.175
OmniMath	.321	.547	+.208
OlympiadBench	.324	.516	+.228

G.2 Controlled-to-Natural Confidence Transfer

The confidence function is transferred from controlled data without ProcessBench calibration. Using the controlled threshold $\tau = .775$, accepted-case exact localization is .512/.391/.346/.429 on GSM8K/MATH/OmniMath/OlympiadBench, with coverage .396/.332/.141/.138. These results are reported as a transfer diagnostic: the controlled reliability score retains useful ordering information when applied to natural traces.

Table 15: Transferred controlled confidence on natural ProcessBench traces. The threshold is fixed before natural evaluation.

Domain	Accepted exact	Coverage
GSM8K	.512	.396
MATH	.391	.332
OmniMath	.346	.141
OlympiadBench	.429	.138

G.3 Cross-Scale Verification Test

A separate Qwen3-32B GSM8K stress test uses the same verification protocol on the first 400 ProcessBench traces. Hidden-state localization reaches .575 exact and .783 within one step. Applying the same controlled threshold accepts 44.0% of error traces at .692 exact accuracy. On the 170 errors with an observable predecessor, hidden-state exact localization reaches .700. This cross-scale test supports the persistence of verification-conditioned boundary structure within the Qwen family.

Table 16: Qwen3-32B verification-conditioned natural localization on ProcessBench GSM8K.

Population	Exact	± 1	Selective exact / coverage
All annotated errors	.575	.783	.692/.440
Observable inter-step errors	.700	.859	.829/.447

G.4 Generator-Family Decomposition

ProcessBench GSM8K contains traces from several generator families while Qwen3-8B remains fixed as the inspector. Verification-conditioned exact localization is .416 for Qwen2 traces, .397 for Qwen2.5 traces, .511 for Llama traces, and .481 for Meta-Llama traces; the corresponding within-one-step accuracies are .753/.690/.778/.778. This decomposition shows that the verification-conditioned signal is not confined to one trace generator.

G.5 Step Alignment and Observability

For every inter-step boundary $t > 1$, the verification score compares the prefix ending at $t - 1$ with the prefix ending at t. The score is therefore aligned with the newly appended reasoning step and does not introduce an off-by-one shift. First-step errors have no preceding reasoning prefix and are treated as a separate observability condition in the implementation. The main natural conclusions are supported by the full annotated population and by the observable inter-step subset reported above.

Table 17: Verification-conditioned localization by ProcessBench GSM8K trace generator.

Generator family	n	Exact	± 1
Qwen2	77	.416	.753
Qwen2.5	58	.397	.690
Llama	45	.511	.778
Meta-Llama	27	.481	.778

H Claim–Evidence Map

The appendix is organized to make each main-text claim traceable to an explicit evidence block.

- *Controlled perturbations produce reproducible localizable transitions.* Evidence: fixed-configuration results, best-configuration summaries, held-out configuration selection, corruption-type decompositions, and the three-seed central campaign.
- *Depth structure generalizes prospectively.* Evidence: Qwen3-32B depth profile and the two pre-specified directional predictions in Appendix F.
- *Inspection computation changes natural boundary strength.* Evidence: the four-domain verification-conditioned study, the generator-family decomposition, and the Qwen3-32B cross-scale stress test in Appendix G.
- *Reliability information transfers across regimes.* Evidence: controlled-threshold selective results in Table 15 and the held-out controlled selective operating points in the generated tables.
- *Localized boundaries can support downstream regeneration.* Evidence: controlled targeted-regeneration results and the same-model passive-to-verification repair comparison in Appendix Q.2.

I Detector Formalization and Algorithms

Our formulation is training-free: it treats the model's hidden-state sequence as a trajectory and applies a scalar change-point rule, consistent with prior activation-localization work that identifies internal computation by comparing transformer representations (Meng et al., 2022).

Algorithm 1 Reasoning Change-Point Detection (per trajectory)

Require: Trace $\tau = (s_1, \ldots, s_K)$, model $\mathcal{M}$, relative depth set $\mathcal{D}$, signal function g, detector $\mathcal{A}$
Ensure: Predicted first-failure step $\hat{t}$
$H \leftarrow \emptyset$
for $d \in \mathcal{D}$ **do**
  $h_1^{(d)}, \ldots, h_K^{(d)} \leftarrow \textsc{ExtractStates}(\mathcal{M}, \tau, d)$ {one forward pass}
end for
for $d \in \mathcal{D}$ **do**
  $I_t^{(d)} \leftarrow g(h_t^{(d)}, h_{t-1}^{(d)})$ for $t = 2, \ldots, K$ {innovation series}
  $\hat{t}^{(d)} \leftarrow \mathcal{A}(I_2^{(d)}, \ldots, I_K^{(d)})$ {argmax / z-score / CUSUM / PELT}
end for
return $\hat{t} \leftarrow$ aggregate$(\{\hat{t}^{(d)}\}_{d \in \mathcal{D}})$ {single-depth or fusion}

The three geometric signals are:

$$I_t^{\cos} = 1 - \frac{\langle h_t, h_{t-1} \rangle}{\|h_t\| \, \|h_{t-1}\|}, \tag{1}$$

$$I_t^{\ell_2} = \frac{\|h_t - h_{t-1}\|_2}{\|h_{t-1}\|_2 + \epsilon}, \tag{2}$$

$$I_t^{\mathrm{curv}} = 1 - \cos\big(h_t - h_{t-1},\ h_{t-1} - h_{t-2}\big). \tag{3}$$

Cosine innovation measures angular change regardless of magnitude; normalized ℓ_2 displacement scales the step displacement by the norm of the preceding hidden state; curvature measures the change in trajectory direction. The cosine operations normalize internally through their denominators; the hidden vectors themselves are not pre-normalized before these signals are computed.

The four detectors operate on the scalar innovation series $(I_2, \ldots, I_K)$:

- **Argmax**: $\hat{t} = \arg\max_{t \in \mathcal{C}(\tau)} I_t$. The simplest local rule; it detects the largest eligible innovation directly.
- **Z-score**: standardize the eligible innovation values by their within-trace mean and standard deviation, then return the eligible index with maximum standardized value. Because this transformation is monotone within a trace, it is equivalent to argmax unless ties or numerical effects intervene.
- **CUSUM**: on the eligible values, compute $S_j = \sum_{i \le j}(I_i - \bar{I})$, shift the cumulative sequence by its minimum, and return the eligible index at the maximum shifted cumulative value. The implementation therefore targets an upward cumulative departure rather than the maximum absolute excursion.
- **PELT** (Killick et al., 2012): apply `ruptures.Pelt` with RBF cost, `min_size=1`, `jump=1`, and penalty 1.0 to the eligible innovation sequence; if multiple segments are returned, the first segment end is used as the change location. PELT is included as a classical segmentation reference for the short-series regime.

J Statistical Methods

Bootstrap confidence intervals. For a proportion $p = k/n$ (exact or ± 1 localization accuracy), we compute percentile bootstrap 95% confidence intervals by resampling trajectories with replacement $B = 10{,}000$ times and taking the 2.5th and 97.5th percentiles of the bootstrap distribution. For paired contrasts (e.g., change-point vs. random repair accuracy), we resample trajectory pairs and compute the difference in accuracy for each bootstrap replicate, yielding a confidence interval for the resulting contrast estimate under the stated trajectory-pair resampling procedure.

Permutation tests. For paired contrasts we use a permutation test: under the null of no difference, the labels (method A vs. method B) are exchangeable within each trajectory. We permute labels $10{,}000$ times and compute the proportion of permutations yielding a difference at least as extreme as the observed difference. This gives an exact (non-asymptotic) p-value without distributional assumptions under the stated exchangeability null.

McNemar test. For comparing two classifiers (configurations) on the same set of trajectories, we use McNemar's test on the discordant pairs (trajectories where the two configurations disagree). We report the exact binomial p-value.

Table 18: Statistical procedures used in the main and supplementary analyses.

Analysis	Procedure	Setting
Accuracy interval	Trajectory bootstrap	$B = 10{,}000$, percentile 95% CI
Paired repair contrast	Paired bootstrap / permutation	$B = 10{,}000$
Paired configuration comparison	Exact McNemar test	Discordant pairs
Selective threshold	One-sided bootstrap lower bound	target accepted exact $\geq .80$

Per-trajectory random baseline. Random localization is defined over the candidate set used by each regime. In controlled experiments the candidate set is the eligible inter-step positions $\{2, \ldots, K\}$; in ProcessBench natural evaluation it is all K annotated step positions so that first-step errors remain in the population. Exact chance accuracy for a trace is the reciprocal of the corresponding candidate-set size. We average this quantity over traces rather than using a single

global constant because trace lengths vary. When ± 1 random accuracy is reported, its per-trace value is computed from the number of candidate positions lying within one step of that trace's target, so boundary targets are handled explicitly.

K Error Propagation Analysis

An important property of the controlled corruption protocol is that not all injected errors propagate to a wrong final answer. Across models and benchmarks, 36–60% of corrupted traces produce an incorrect final answer (propagate), while the remainder recover to a correct answer despite the injected error. This variation is informative: a corrupted trace that recovers demonstrates that the model can sometimes absorb a local error without it cascading. Table 19 reports the propagation rate per model on GSM8K.

Table 19: Error propagation rate (fraction of corrupted traces whose final answer becomes incorrect) on GSM8K, per model.

Model	n	prop. rate	mean K
Llama-3-8B	991	.597	4.2
Mistral-7B	691	.554	4.3
Qwen3-4B	1131	.472	4.2
Qwen3-8B	984	.441	4.5
Phi-4-mini	1048	.434	4.4

We analyze localization accuracy conditioned on propagation status. Across the main controlled settings, localization on propagating corruptions remains substantial, indicating that the transition signal is not restricted to errors that are later absorbed by the model.

The propagation rate varies sharply by corruption type (Table 20). Arithmetic corruptions propagate to a wrong final answer in 84% of cases, whereas quantity swaps propagate in 14%. Together with the localization decomposition in Appendix T.3, this shows that corruption type shapes both the local boundary signature and the downstream trajectory, while these two properties need not vary identically.

Table 20: Propagation rate by corruption type on GSM8K (all models pooled).

Corruption type	n	prop. rate
Arithmetic perturbation	2461	.842
Quantity swap	2384	.137

L Configuration Selection Robustness

We evaluate split stability with a held-out selection protocol: each model–benchmark population is divided into a selection half and a disjoint evaluation half, the strongest configuration is chosen on the selection half only, and the selected configuration is then evaluated unchanged on unseen pairs. Table 21 reports the resulting evaluation-half accuracy.

The selected configurations preserve strong localization on the disjoint evaluation halves across all twelve model–benchmark cells. The same controlled campaign also yields high-coverage reliability operating points for Qwen3-8B and Mistral-7B after threshold selection on calibration data.

M Broader Impact and Ethical Considerations

Our work studies internal signals associated with reasoning errors, with applications to debugging, adaptive verification, and selective regeneration. Verification conditioning re-evaluates accumulated prefixes and therefore increases inference cost relative to a single-pass generation. The transferred

Table 21: Held-out selected configurations evaluated on disjoint pairs.

Model	Benchmark	n_{eval}	Exact	± 1
Qwen3-4B	GSM8K	250	.798	.933
Qwen3-4B	MATH-500	102	.650	.854
Qwen3-8B	GSM8K	250	.758	.877
Qwen3-8B	MATH-500	101	.772	.901
Llama-3-8B	GSM8K	250	.711	.880
Llama-3-8B	MATH-500	38	.538	.795
Mistral-7B	GSM8K	250	.855	.978
Mistral-7B	MATH-500	30	.800	.900
Phi-4-mini	GSM8K	250	.595	.861
Phi-4-mini	MATH-500	74	.527	.770
DS-R1-7B	GSM8K	159	.818	.962
DS-R1-7B	MATH-500	79	.762	.875

Table 22: Held-out controlled selective operating points.

Model	τ	Coverage	Accepted exact	One-sided 95% LB
Qwen3-8B	.775	.864	.836	.806
Mistral-7B	.850	.868	.851	.817

confidence score is used as a selective operating signal rather than as a replacement for task-specific safety validation.

We release code and processed signals to support reproducibility. The ProcessBench dataset (Zheng et al., 2025) is released under Apache-2.0. The model weights used (Qwen3, Llama-3, Mistral, Phi-4-mini, DeepSeek-R1-Distill) are all publicly released under permissive licenses.

N Reproducibility Checklist

- **Datasets**: GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), and the GSM8K, MATH, OmniMath, and OlympiadBench domains of ProcessBench (Zheng et al., 2025). All are publicly available.
- **Models**: Qwen3-4B/8B/32B, Llama-3-8B-Instruct, Mistral-7B-Instruct, Phi-4-mini-instruct, DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025). All publicly available.
- **Hyperparameters**: greedy decoding (temperature 0), bf16, relative depths $\{25,50,75,100\}\%$, last-token and mean-pool representations, cosine/ℓ_2/curvature signals, argmax/z-score/CUSUM/PELT detectors. No training or gradient updates.
- **Compute**: controlled signal extraction runs on a single RTX 3090 GPU per model (Qwen3-32B uses multiple RTX 3090s). Verification-conditioned natural localization re-evaluates each reasoning prefix under the frozen verification prompt and therefore requires multiple forward passes per trace.
- **Code**: the full pipeline (generation, corruption, signal extraction, detection, repair, evaluation) is released at submission time.
- **Randomness**: greedy model decoding and controlled corruption selection are deterministic. Explicitly randomized procedures, including random-cut repair controls and bootstrap/permutation analyses, use fixed seeds recorded in the result artifacts ($B = 10{,}000$ resamples where applicable).
- **Frozen 32B rules**: the two prospective rules are recorded in a hashed file before any 32B results; the hash is recorded at submission time.

O Detector Sensitivity Analysis

Eligible-step restriction. Controlled experiments restrict candidate change points to steps $2, \ldots, K$, because the first step has no preceding reasoning step and the corruption protocol never injects at step 1. The supervised PRM baseline uses the same controlled eligible set. Natural ProcessBench evaluation retains the full annotated population, while inter-step analyses additionally report the subset with an observable predecessor. Appendix G documents the alignment convention and the corresponding inter-step results.

Injection-step distribution. The injection step $t^\star$ is selected deterministically from reasoning steps 2 through K, with candidates ordered by distance to the trajectory midpoint and then by step index. Because GSM8K traces are short (mean $K \approx 4.3$) and corruption feasibility depends on step content, the realized distribution is concentrated near the beginning of the trace. Table 23 reports the pooled distribution across corrupted GSM8K trajectories.

Table 23: Injection-step distribution across corrupted GSM8K trajectories (all models pooled, $n = 4845$).

$t^\star$	2	3	4	5	≥ 6
% of traces	62.8	25.1	7.1	2.3	2.7

Signal normalization. The cosine innovation (Eq. 1) is scale-invariant by construction. The normalized ℓ_2 displacement (Eq. 2) divides each step displacement by the norm of the preceding hidden state. This controls for broad scale differences in activation magnitude across layers and model widths while preserving local relative movement; the normalization is part of the predefined signal rather than a tuned performance trick.

P Notation and Definitions

Table 24: Summary of notation used throughout the paper.

Symbol	Definition
$\tau = (s_1, \ldots, s_K, a)$	Reasoning trace with K steps and final answer a
$h_t^{(\ell)}$	Hidden state of step t at layer ℓ
$t^\star$	True first-error step (injected in controlled; annotated in natural)
$\hat{t}$	Predicted failure step
I_t	Innovation signal at step t
K	Number of reasoning steps
$\mathcal{C}(\tau)$	Candidate localization positions for trace τ
$\mathcal{D}$	Set of relative depths $\{25, 50, 75, 100\}\%$
E	Exact localization accuracy $\mathbb{1}[\hat{t} = t^\star]$
± 1	Localization within one step $\mathbb{1}[\lvert\hat{t} - t^\star\rvert \leq 1]$
$A(\tau)$	Acceptance indicator for selective localization

Eligible steps. For controlled localization, eligible change points run from reasoning step 2 through step K, because the injected failure is never placed at step 1. Natural ProcessBench evaluation retains all annotated error traces in the primary population. We additionally report observable inter-step subsets when the consecutive-prefix score requires a predecessor.

Relative depth. For a model with L layers, relative depth $d \in \{25, 50, 75, 100\}\%$ maps to the implementation index $\mathrm{round}(dL/100)$ clipped to $[1, L]$. For example, Qwen3-8B ($L = 36$) uses layers 9, 18, 27, 36 and Qwen3-32B ($L = 64$) uses 16, 32, 48, 64. This normalization ensures that the same relative depths are compared across models of different sizes.

Q Downstream Intervention Analyses

The downstream experiments test whether localized boundaries can be converted into useful reasoning-control decisions. We report controlled targeted-regeneration contrasts, followed by a same-model natural study that keeps the generator, inspector, and repair checkpoint fixed.

Q.1 Controlled Targeted-Regeneration Contrasts

On the controlled GSM8K intervention set, RCP cuts improve final-answer accuracy over the original corrupted trajectory for Qwen3-8B and Mistral-7B, and also outperform paired random-cut regeneration in the corresponding audit.

Table 25: Positive paired contrasts for controlled GSM8K targeted regeneration.

Model	Contrast	n	Accuracy gain	95% CI
Mistral-7B	RCP vs. random cut	150	+.261	[.167,.353]
Mistral-7B	RCP vs. original	150	+.167	[.067,.267]
Qwen3-8B	RCP vs. random cut	150	+.146	[.040,.247]

Q.2 Same-Model Natural Repair Without First-Error Labels

We run a label-free same-model study using 276 parseable incorrect GSM8K traces produced greedily by Qwen3-8B itself. The generator, hidden-state inspector, and repair model are therefore the same frozen checkpoint. The passive detector uses final-layer consecutive-state cosine change on the original generation trajectory, while the verification detector uses the same native-chat verification instruction and consecutive-prefix cosine change as the ProcessBench experiment. Neither cut policy is fitted with first-error labels.

For a predicted cut k, the repair prompt preserves reasoning steps through k and asks Qwen3-8B to reconsider the solution from the next step onward. All cut-based methods use the same prompt, greedy decoding, and a 384-token budget.

Table 26: Same-model natural repair with Qwen3-8B under a fixed prefix-preserving repair prompt.

Cut policy	Accuracy	Gain	95% CI on gain	Wins/Losses
Passive hidden-state cut	.326	–	–	–
Verification-conditioned cut	.442	+.116	[.062,.170]	46/14

The verification-conditioned cut policy improves repair accuracy by 11.6 percentage points under identical decoding conditions, connecting the stronger verification-conditioned boundary signal to a stronger prefix-preserving intervention.

R Extended Protocol and Reporting Details

R.1 Generation Prompt Template

Controlled traces use the instruction “Solve the following math problem step by step. End with ‘Final Answer: ¡answer¿.’.”, followed by the problem statement. The same parsing rules are applied across models: numbered reasoning steps are extracted from `Step k:` markers and the final answer is read from the `Final Answer:` line. Distilled-R1 `<think>` content is excluded from step parsing.

R.2 Verification Prompt Semantics

The verification-conditioned computation presents the original problem together with the accumulated reasoning prefix and asks the frozen model whether the reasoning shown so far contains a mathematical or logical error. The prompt family, chat formatting, hidden-state readout location, and

change rule are fixed across all four ProcessBench domains. The appendix reports the protocol at the semantic level because the released artifact bundle contains the exact prompt strings used by the experiments.

R.3 Repair Prompt Template

For self-reflection references, the corrupted trace is followed by the instruction "Review your reasoning above. If any step is wrong, correct it and continue. End with 'Final Answer: ¡answer¿'." For prefix-preserving RCP repair, the selected prefix is retained and the model is asked to continue from the next reasoning step with the same frozen checkpoint.

R.4 Computation and Storage

Controlled hidden-state extraction records the selected relative depths in a single forward pass. Verification-conditioned natural localization performs one frozen-model evaluation per accumulated prefix. This makes verification more expensive than passive extraction but also exposes a distinct computational representation of the same reasoning trace. The artifact bundle stores per-trajectory predictions and the scalar summaries required to regenerate all reported appendix tables.

S Adaptive Reasoning Control: Downstream Directions

The experiments motivate several downstream uses of boundary reliability. First, a system can preserve a high-confidence prefix and regenerate only the suffix after the localized boundary. Second, a low-cost generation-time monitor can decide whether an additional verification-conditioned pass is worthwhile. Third, confidence can be used as a routing signal for allocating more inference compute or human review to trajectories whose boundary evidence warrants it. These are prospective system designs rather than additional empirical claims; they follow from the observed ability to strengthen and selectively rank first-error boundary evidence.

T Extended Configuration and Robustness Tables

The following generated tables provide the exhaustive configuration landscapes and additional robustness decompositions that complement the compact analyses above. They include full last-token and mean-pooling grids, per-corruption-type results, corrected cross-benchmark summaries, the three-seed campaign, and the machine-readable artifact index.

Table 27: Supplementary evidence inventory.

Evidence axis	Supplementary coverage
Configuration landscape	Full last-token and mean-pooling depth × signal × detector grids
Corruption structure	Per-type localization, propagation rates, injection-step distribution
Generalization	Held-out selection, cross-benchmark summary, prospective Qwen3-32B depth test
Natural localization	Four domains, controlled-to-natural confidence transfer, generator-family and cross-scale analyses
Stability	Frozen three-seed campaign with depth profiles and validation records
Intervention	Controlled targeted regeneration and same-model natural prefix-preserving repair
Reproducibility	Exact protocol descriptions and machine-readable artifact index

T.1 Controlled Configuration Landscapes

Tables 28–29 report the full depth-by-signal-by-detector grid at the last-token representation for each model. Each cell is exact localization accuracy. These tables expose the configuration landscape underlying the selected results in the main text. They are included for reproducibility rather than as separate headline comparisons.

Table 28: Full localization grid (GSM8K, last-token representation). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Qwen3-4B	100%	cos	**0.799**	**0.799**	0.706	0.391
	100%	cur	0.158	0.158	0.153	0.254
	100%	l2	0.715	0.715	0.634	0.409
	25%	cos	0.466	0.466	0.426	0.279
	25%	cur	0.087	0.087	0.082	0.278
	25%	l2	0.528	0.528	0.454	0.259
	50%	cos	0.466	0.466	0.420	0.250
	50%	cur	0.101	0.101	0.098	0.240
	50%	l2	0.489	0.489	0.430	0.219
	75%	cos	0.554	0.554	0.505	0.266
	75%	cur	0.110	0.110	0.111	0.224
	75%	l2	0.539	0.539	0.496	0.305
Qwen3-8B	100%	cos	**0.770**	**0.770**	0.702	0.307
	100%	cur	0.062	0.062	0.062	0.169
	100%	l2	0.717	0.717	0.651	0.356
	25%	cos	0.555	0.555	0.505	0.242
	25%	cur	0.035	0.035	0.029	0.333
	25%	l2	0.762	0.762	0.589	0.226
	50%	cos	0.623	0.623	0.561	0.236
	50%	cur	0.052	0.052	0.051	0.247
	50%	l2	0.654	0.654	0.565	0.207
	75%	cos	0.663	0.663	0.619	0.162
	75%	cur	0.065	0.065	0.061	0.222
	75%	l2	0.666	0.666	0.614	0.194
Llama-3-8B	100%	cos	**0.710**	**0.710**	0.681	0.410
	100%	cur	0.118	0.118	0.107	0.109
	100%	l2	0.700	0.700	0.666	0.376
	25%	cos	0.364	0.364	0.359	0.249
	25%	cur	0.093	0.093	0.097	0.167
	25%	l2	0.327	0.327	0.353	0.267
	50%	cos	0.368	0.368	0.394	0.292
	50%	cur	0.097	0.097	0.092	0.182
	50%	l2	0.376	0.376	0.394	0.304
	75%	cos	0.415	0.415	0.424	0.264
	75%	cur	0.105	0.105	0.085	0.235
	75%	l2	0.422	0.422	0.424	0.286

Table 28: Full localization grid (GSM8K, last-token representation; continued). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Mistral-7B	100%	cos	0.792	0.792	0.870	0.644
	100%	cur	0.288	0.288	0.267	0.410
	100%	l2	0.816	0.816	0.848	0.601
	25%	cos	0.700	0.700	0.810	0.681
	25%	cur	0.256	0.256	0.253	0.267
	25%	l2	**0.875**	**0.875**	0.863	0.637
	50%	cos	0.716	0.716	0.756	0.545
	50%	cur	0.193	0.193	0.173	0.161
	50%	l2	0.857	0.857	0.787	0.444
	75%	cos	0.677	0.677	0.738	0.432
	75%	cur	0.244	0.244	0.223	0.340
	75%	l2	0.721	0.721	0.741	0.406
Phi-4-mini	100%	cos	0.487	0.487	0.499	0.226
	100%	cur	0.141	0.141	0.131	0.305
	100%	l2	0.484	0.484	0.499	0.253
	25%	cos	0.377	0.377	0.364	0.253
	25%	cur	0.114	0.114	0.110	0.281
	25%	l2	0.365	0.365	0.360	0.233
	50%	cos	0.331	0.331	0.304	0.242
	50%	cur	0.119	0.119	0.111	0.337
	50%	l2	0.290	0.290	0.276	0.245
	75%	cos	0.529	0.529	0.518	0.322
	75%	cur	0.119	0.119	0.115	0.236
	75%	l2	**0.530**	**0.530**	0.521	0.352
DS-R1-7B	100%	cos	0.711	0.711	0.770	0.560
	100%	cur	0.098	0.098	0.088	0.200
	100%	l2	0.777	0.777	**0.802**	0.614
	25%	cos	0.525	0.525	0.544	0.258
	25%	cur	0.107	0.107	0.107	0.210
	25%	l2	0.528	0.528	0.528	0.273
	50%	cos	0.497	0.497	0.487	0.242
	50%	cur	0.126	0.126	0.148	0.273
	50%	l2	0.469	0.469	0.462	0.267
	75%	cos	0.585	0.585	0.607	0.229
	75%	cur	0.135	0.135	0.129	0.357
	75%	l2	0.563	0.563	0.616	0.391

Table 29: Full localization grid (MATH-500, last-token representation). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Qwen3-4B	100%	cos	**0.624**	**0.624**	0.546	0.218
	100%	cur	0.180	0.180	0.215	0.207
	100%	l2	0.541	0.541	0.439	0.281
	25%	cos	0.376	0.376	0.356	0.286
	25%	cur	0.180	0.180	0.166	0.154
	25%	l2	0.385	0.385	0.361	0.318
	50%	cos	0.415	0.415	0.390	0.212
	50%	cur	0.176	0.176	0.176	0.367
	50%	l2	0.390	0.390	0.371	0.226
	75%	cos	0.444	0.444	0.429	0.210
	75%	cur	0.156	0.156	0.146	0.200
	75%	l2	0.419	0.419	0.434	0.218
Qwen3-8B	100%	cos	**0.767**	**0.767**	0.713	0.463
	100%	cur	0.129	0.129	0.134	0.460
	100%	l2	0.673	0.673	0.634	0.431
	25%	cos	0.411	0.411	0.381	0.351
	25%	cur	0.148	0.148	0.124	0.382
	25%	l2	0.505	0.505	0.396	0.343
	50%	cos	0.431	0.431	0.342	0.319
	50%	cur	0.134	0.134	0.144	0.302
	50%	l2	0.431	0.431	0.356	0.243
	75%	cos	0.535	0.535	0.455	0.271
	75%	cur	0.119	0.119	0.148	0.357
	75%	l2	0.455	0.455	0.451	0.290
Llama-3-8B	100%	cos	**0.545**	**0.545**	0.506	0.320
	100%	cur	0.156	0.156	0.104	0.182
	100%	l2	**0.545**	**0.545**	0.506	0.320
	25%	cos	0.221	0.221	0.312	0.304
	25%	cur	0.182	0.182	0.156	0.133
	25%	l2	0.221	0.221	0.234	0.318
	50%	cos	0.299	0.299	0.286	0.375
	50%	cur	0.156	0.156	0.169	0.286
	50%	l2	0.299	0.299	0.299	0.292
	75%	cos	0.377	0.377	0.312	0.333
	75%	cur	0.182	0.182	0.143	0.267
	75%	l2	0.260	0.260	0.325	0.250

Table 29: Full localization grid (MATH-500, last-token representation; continued). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Mistral-7B	100%	cos	0.680	0.680	0.880	0.368
	100%	cur	0.460	0.460	0.500	0.600
	100%	l2	0.780	0.780	**0.900**	0.389
	25%	cos	0.520	0.520	0.840	0.562
	25%	cur	0.460	0.460	0.440	0.500
	25%	l2	0.760	0.760	0.860	0.529
	50%	cos	0.700	0.700	0.700	0.312
	50%	cur	0.360	0.360	0.300	0.250
	50%	l2	0.740	0.740	0.700	0.462
	75%	cos	0.580	0.580	0.660	0.500
	75%	cur	0.360	0.360	0.360	0.143
	75%	l2	0.620	0.620	0.600	0.467
Phi-4-mini	100%	cos	0.473	0.473	**0.507**	0.200
	100%	cur	0.162	0.162	0.182	0.216
	100%	l2	0.460	0.460	0.460	0.237
	25%	cos	0.284	0.284	0.284	0.282
	25%	cur	0.176	0.176	0.162	0.167
	25%	l2	0.270	0.270	0.284	0.284
	50%	cos	0.304	0.304	0.304	0.261
	50%	cur	0.196	0.196	0.182	0.300
	50%	l2	0.264	0.264	0.318	0.231
	75%	cos	0.405	0.405	0.432	0.324
	75%	cur	0.230	0.230	0.203	0.219
	75%	l2	0.392	0.392	0.419	0.282
DS-R1-7B	100%	cos	0.692	0.692	**0.774**	0.513
	100%	cur	0.145	0.145	0.101	0.273
	100%	l2	0.742	0.742	0.748	0.500
	25%	cos	0.358	0.358	0.390	0.167
	25%	cur	0.132	0.132	0.132	0.182
	25%	l2	0.365	0.365	0.409	0.237
	50%	cos	0.403	0.403	0.390	0.200
	50%	cur	0.157	0.157	0.176	0.350
	50%	l2	0.390	0.390	0.409	0.220
	75%	cos	0.509	0.509	0.566	0.321
	75%	cur	0.113	0.113	0.113	0.200
	75%	l2	0.509	0.509	0.560	0.214

T.2 Mean-Pooling Configuration Landscapes

Tables 30–31 repeat the full grid at the mean-pooling representation. Mean pooling underperforms last-token states for most models, with Llama-3-8B and Phi-4-mini as the main exceptions where mean pooling is competitive or better. This confirms that the choice of step representation is a genuine per-model configuration dimension.

Table 30: Mean-pooling localization grid (GSM8K). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Qwen3-4B	100%	cos	**0.392**	**0.392**	0.374	0.365
	100%	cur	0.155	0.155	0.164	0.305
	100%	l2	0.382	0.382	0.370	0.369
	25%	cos	0.340	0.340	0.380	0.269
	25%	cur	0.107	0.107	0.086	0.224
	25%	l2	0.338	0.338	0.383	0.254
	50%	cos	0.321	0.321	0.355	0.267
	50%	cur	0.135	0.135	0.121	0.200
	50%	l2	0.290	0.290	0.320	0.285
	75%	cos	0.346	0.346	0.366	0.287
	75%	cur	0.143	0.143	0.149	0.304
	75%	l2	0.327	0.327	0.344	0.303
Qwen3-8B	100%	cos	**0.553**	**0.553**	0.507	0.380
	100%	cur	0.138	0.138	0.178	0.365
	100%	l2	0.529	0.529	0.469	0.345
	25%	cos	0.415	0.415	0.467	0.194
	25%	cur	0.105	0.105	0.100	0.160
	25%	l2	0.415	0.415	0.465	0.218
	50%	cos	0.397	0.397	0.441	0.213
	50%	cur	0.126	0.126	0.130	0.192
	50%	l2	0.377	0.377	0.426	0.174
	75%	cos	0.422	0.422	0.432	0.258
	75%	cur	0.127	0.127	0.129	0.224
	75%	l2	0.380	0.380	0.402	0.255
Llama-3-8B	100%	cos	0.504	0.504	**0.552**	0.383
	100%	cur	0.091	0.091	0.106	0.161
	100%	l2	0.541	0.541	0.546	0.363
	25%	cos	0.366	0.366	0.418	0.246
	25%	cur	0.066	0.066	0.067	0.333
	25%	l2	0.339	0.339	0.389	0.282
	50%	cos	0.305	0.305	0.328	0.361
	50%	cur	0.090	0.090	0.080	0.210
	50%	l2	0.240	0.240	0.264	0.347
	75%	cos	0.358	0.358	0.392	0.245
	75%	cur	0.080	0.080	0.080	0.200
	75%	l2	0.349	0.349	0.368	0.271

T.3 Corruption-Type Breakdown: Full Results

Table 32 reports per-model, per-corruption-type exact localization at each model's selected configuration. Quantity swaps are especially localizable for the Qwen models, while arithmetic and quantity-swap performance is closer for Llama-3-8B and Mistral-7B. Combined with the propagation statistics in Appendix K, this decomposition shows that local boundary salience and downstream propagation provide complementary views of corruption behavior.

T.4 Cross-Benchmark Consistency

GSM8K and MATH-500 differ in difficulty, step count, and answer format. Table 33 compares the strongest selected configuration on each benchmark. The most stable structural pattern is depth:

Table 30: Mean-pooling localization grid (GSM8K; continued). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Mistral-7B	100%	cos	0.365	0.365	0.377	0.340
	100%	cur	0.093	0.093	0.095	0.302
	100%	l2	0.354	0.354	0.372	0.326
	25%	cos	0.342	0.342	**0.389**	0.294
	25%	cur	0.080	0.080	0.077	0.310
	25%	l2	0.330	0.330	0.379	0.324
	50%	cos	0.305	0.305	0.312	0.368
	50%	cur	0.097	0.097	0.087	0.265
	50%	l2	0.258	0.258	0.262	0.295
	75%	cos	0.333	0.333	0.324	0.359
	75%	cur	0.092	0.092	0.100	0.213
	75%	l2	0.306	0.306	0.300	0.337
Phi-4-mini	100%	cos	0.552	0.552	0.581	0.368
	100%	cur	0.101	0.101	0.126	0.256
	100%	l2	0.601	0.601	**0.616**	0.429
	25%	cos	0.399	0.399	0.446	0.271
	25%	cur	0.082	0.082	0.076	0.265
	25%	l2	0.386	0.386	0.432	0.274
	50%	cos	0.323	0.323	0.335	0.315
	50%	cur	0.106	0.106	0.096	0.311
	50%	l2	0.290	0.290	0.315	0.289
	75%	cos	0.425	0.425	0.465	0.302
	75%	cur	0.103	0.103	0.108	0.247
	75%	l2	0.407	0.407	0.453	0.333
DS-R1-7B	100%	cos	0.616	0.616	0.660	0.481
	100%	cur	0.157	0.157	0.141	0.111
	100%	l2	0.676	0.676	**0.679**	0.490
	25%	cos	0.531	0.531	0.598	0.276
	25%	cur	0.101	0.101	0.098	0.316
	25%	l2	0.541	0.541	0.604	0.268
	50%	cos	0.462	0.462	0.513	0.235
	50%	cur	0.116	0.116	0.094	0.167
	50%	l2	0.421	0.421	0.440	0.200
	75%	cos	0.572	0.572	0.607	0.260
	75%	cur	0.123	0.123	0.126	0.214
	75%	l2	0.497	0.497	0.528	0.208

Table 31: Mean-pooling localization grid (MATH-500). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Qwen3-4B	100%	cos	**0.376**	**0.376**	0.371	0.243
	100%	cur	0.176	0.176	0.180	0.192
	100%	l2	0.307	0.307	0.307	0.343
	25%	cos	0.278	0.278	0.327	0.343
	25%	cur	0.190	0.190	0.180	0.360
	25%	l2	0.278	0.278	0.342	0.349
	50%	cos	0.298	0.298	0.332	0.298
	50%	cur	0.185	0.185	0.176	0.357
	50%	l2	0.307	0.307	0.351	0.321
	75%	cos	0.317	0.317	0.337	0.292
	75%	cur	0.195	0.195	0.171	0.333
	75%	l2	0.337	0.337	0.332	0.318
Qwen3-8B	100%	cos	**0.599**	**0.599**	0.485	0.361
	100%	cur	0.183	0.183	0.213	0.366
	100%	l2	0.510	0.510	0.416	0.284
	25%	cos	0.247	0.247	0.337	0.231
	25%	cur	0.168	0.168	0.193	0.297
	25%	l2	0.253	0.253	0.337	0.230
	50%	cos	0.267	0.267	0.282	0.247
	50%	cur	0.134	0.134	0.158	0.326
	50%	l2	0.267	0.267	0.287	0.263
	75%	cos	0.282	0.282	0.297	0.236
	75%	cur	0.163	0.163	0.168	0.273
	75%	l2	0.272	0.272	0.297	0.258
Llama-3-8B	100%	cos	0.429	0.429	**0.506**	0.348
	100%	cur	0.195	0.195	0.208	0.357
	100%	l2	0.480	0.480	0.455	0.286
	25%	cos	0.325	0.325	0.312	0.217
	25%	cur	0.208	0.208	0.156	0.214
	25%	l2	0.299	0.299	0.325	0.263
	50%	cos	0.325	0.325	0.247	0.250
	50%	cur	0.221	0.221	0.156	0.167
	50%	l2	0.273	0.273	0.195	0.222
	75%	cos	0.351	0.351	0.325	0.316
	75%	cur	0.208	0.208	0.169	0.250
	75%	l2	0.325	0.325	0.312	0.250

Table 31: Mean-pooling localization grid (MATH-500; continued). Exact accuracy; best per model in bold.

Model	Depth	Signal	arg	zsc	cus	pelt
Mistral-7B	100%	cos	0.260	0.260	0.300	0.235
	100%	cur	0.180	0.180	0.140	0.333
	100%	l2	0.260	0.260	0.320	0.176
	25%	cos	0.200	0.200	0.300	0.077
	25%	cur	0.140	0.140	0.160	**0.500**
	25%	l2	0.180	0.180	0.280	0.231
	50%	cos	0.160	0.160	0.220	0.357
	50%	cur	0.180	0.180	0.100	**0.500**
	50%	l2	0.120	0.120	0.140	0.417
	75%	cos	0.200	0.200	0.220	0.286
	75%	cur	0.180	0.180	0.100	0.250
	75%	l2	0.220	0.220	0.200	0.071
Phi-4-mini	100%	cos	0.318	0.318	0.358	0.329
	100%	cur	0.149	0.149	0.176	0.217
	100%	l2	0.331	0.331	**0.385**	0.358
	25%	cos	0.196	0.196	0.250	0.284
	25%	cur	0.142	0.142	0.122	0.175
	25%	l2	0.209	0.209	0.236	0.304
	50%	cos	0.182	0.182	0.209	0.338
	50%	cur	0.142	0.142	0.142	0.220
	50%	l2	0.169	0.169	0.182	0.349
	75%	cos	0.270	0.270	0.311	0.224
	75%	cur	0.142	0.142	0.135	0.143
	75%	l2	0.257	0.257	0.290	0.242
DS-R1-7B	100%	cos	0.572	0.572	0.685	0.605
	100%	cur	0.189	0.189	0.157	0.267
	100%	l2	0.692	0.692	**0.742**	0.571
	25%	cos	0.327	0.327	0.421	0.225
	25%	cur	0.088	0.088	0.075	0.308
	25%	l2	0.308	0.308	0.365	0.208
	50%	cos	0.321	0.321	0.365	0.256
	50%	cur	0.082	0.082	0.069	0.353
	50%	l2	0.277	0.277	0.283	0.324
	75%	cos	0.403	0.403	0.440	0.293
	75%	cur	0.101	0.101	0.082	0.333
	75%	l2	0.358	0.358	0.403	0.316

Table 32: Per-corruption-type exact localization accuracy at each model's best configuration (with 95% bootstrap CI).

Model	Bench	Type	n	Exact [95% CI]
Qwen3-4B	GSM8K	arithmetic	504	0.728 [0.691, 0.766]
Qwen3-4B	GSM8K	quantity_swap	376	0.891 [0.859, 0.920]
Qwen3-8B	GSM8K	arithmetic	359	0.649 [0.599, 0.696]
Qwen3-8B	GSM8K	quantity_swap	374	0.864 [0.829, 0.898]
Llama-3-8B	GSM8K	arithmetic	398	0.716 [0.671, 0.761]
Llama-3-8B	GSM8K	quantity_swap	245	0.702 [0.645, 0.759]
Mistral-7B	GSM8K	arithmetic	197	0.843 [0.792, 0.893]
Mistral-7B	GSM8K	quantity_swap	216	0.866 [0.819, 0.907]
Phi-4-mini	GSM8K	arithmetic	264	0.549 [0.489, 0.610]
Phi-4-mini	GSM8K	quantity_swap	449	0.621 [0.577, 0.666]
DS-R1-7B	GSM8K	arithmetic	57	0.772 [0.667, 0.877]
DS-R1-7B	GSM8K	quantity_swap	102	0.843 [0.774, 0.912]
Qwen3-4B	MATH-500	arithmetic	41	0.512 [0.366, 0.658]
Qwen3-4B	MATH-500	quantity_swap	62	0.742 [0.629, 0.855]
Qwen3-8B	MATH-500	arithmetic	36	0.722 [0.583, 0.861]
Qwen3-8B	MATH-500	quantity_swap	65	0.800 [0.692, 0.892]
Llama-3-8B	MATH-500	arithmetic	21	0.571 [0.333, 0.762]
Llama-3-8B	MATH-500	quantity_swap	18	0.500 [0.278, 0.722]
Mistral-7B	MATH-500	arithmetic	10	0.900 [0.700, 1.000]
Mistral-7B	MATH-500	quantity_swap	10	0.700 [0.400, 1.000]
Phi-4-mini	MATH-500	arithmetic	30	0.467 [0.300, 0.633]
Phi-4-mini	MATH-500	quantity_swap	44	0.568 [0.432, 0.705]
DS-R1-7B	MATH-500	arithmetic	21	0.619 [0.429, 0.809]
DS-R1-7B	MATH-500	quantity_swap	59	0.814 [0.712, 0.898]

five of six models select the final depth on GSM8K and all six do so on MATH-500. Qwen3-8B is nearly unchanged across benchmarks, while Mistral-7B reaches its strongest selected accuracy on MATH-500. The preferred geometric signal remains cosine or normalized-ℓ_2 across all cells.

Table 33: Best selected configuration and exact accuracy by benchmark. "Cfg." abbreviates depth/representation/signal/detector.

Model	GSM8K best	MATH-500 best	Cfg. (GSM8K / MATH-500)
Qwen3-4B	.799	.624	100/last/cos/arg / 100/last/cos/zsc
Qwen3-8B	.770	.767	100/last/cos/zsc / 100/last/cos/arg
Llama-3-8B	.710	.545	100/last/cos/arg / 100/last/cos/zsc
Mistral-7B	.875	.900	25/last/norm/arg / 100/last/norm/cus
Phi-4-mini	.616	.507	100/last/norm/cus / 100/last/cos/cus
DS-R1-7B	.802	.774	100/last/norm/cus / 100/last/cos/cus

T.5 Three-Seed Central Campaign

The central Qwen3-8B/GSM8K campaign was repeated under a protocol frozen before any seed result existed. All three seeds use the same full-depth, last-token, cosine-innovation, argmax readout.

Table 34: Per-seed exact localization for the frozen Qwen3-8B/GSM8K detector.

Seed	n	Exact	95% bootstrap CI
0	309	.828	[.7864,.8706]
1	319	.809	[.7649,.8495]
2	314	.796	[.7516,.8376]
Pooled	942	.811	–

Table 35: Per-seed depth profile for the frozen central campaign.

Seed	n	A25	A50	A75	A100
0	309	.547	.608	.654	.828
1	319	.580	.624	.639	.809
2	314	.551	.634	.675	.796

The full-depth readout is strongest in all three seeds. Relative to the mean of A25/A50/A75, the A100 advantage is .225/.194/.176 for seeds 0/1/2, with a pooled effect of .206 and a predeclared problem-cluster sign-flip $p = 10^{-4}$.

T.6 Validation Records and Frozen Protocols

For each seed, every stored per-problem value was independently recomputed from the seed tables; all validation checks pass. The prospective Qwen3-32B rules and the three-seed campaign protocol are stored in hashed frozen records archived with the submission.

Table 36: Independent seed-campaign validation.

Seed	Problems	Checks passed	Scope
0	309	10/10	detector and depth readouts
1	319	10/10	detector and depth readouts
2	314	10/10	detector and depth readouts

T.7 Machine-Readable Artifact Index

The reported supplementary numbers are tied to machine-readable artifacts and generator scripts. Table 37 lists the principal sources used for the controlled, seed, prospective, and repair analyses. Hashes are SHA-256 (first 16 hex digits) of the files as submitted. The frozen protocol ledger, seed

Table 37: Machine-readable sources for the appendix evidence (SHA-256, truncated).

Artifact	SHA-256	Covers
results/seed_campaign/frozen_stats.json	2887222a8de91f6e	Frozen 3-seed aggregation (Table 34 and depth-profile records)
results/tables/frozen_seed_protocol.json	0552a540ea6680ac	Frozen protocol ledger with deviation records
results/seed_campaign/ seed0_perproblem_prm.csv	d05c4fc4bb6096f5	Seed-0 per-problem readouts (seeds 1–2 analogous)
results/tables/ frozen_32b_prediction_rule.json	6e791ed61c814ac3	Frozen 32B prospective P1/P2 rules and predictor record
results/tables/eval_32b_prediction.json	7d4f1d29e8eae67e	32B prospective outcome 2/2 (Table 12)
results/tables/fusion_summary.csv	20fc6f329a12bcee	Depth/fusion analysis inputs retained in the artifact bundle
results/tables/repair_audit.csv	cba3232a11bb1fea	Repair contrast bootstrap/permutation inputs (Table 25)
results/tables/paper_per_type_summary.csv	311beea0f8c56d91	Per-corruption-type inputs (Table 32)

aggregations, prospective 32B rules, and table-generation inputs are archived unmodified with the submission so that the supplementary evidence can be regenerated from the recorded artifacts.