When Divergence Emerges, Look Back: Exploring Solution-Domain Topology in Collaborative Decoding

Anonymous authors
Paper under double-blind review

Abstract

Token-level small–large model collaborative decoding has a small language model (SLM) generate most reasoning steps and invokes a large language model (LLM) at local disagreements. Existing token-level correction methods mainly alter the current output, but cannot revoke a committed prefix and its internal state; state repair alone also does not establish whether the repaired state can produce more reliable subsequent reasoning. We therefore formulate small–large model collaboration as a post-injection recovery problem under a finite budget and propose the Solution-Domain Topology Tree. When a disagreement is detected, the system rolls back to an earlier reasoning anchor and uses learnable Cache-to-Cache (C2C) projection to construct a repaired state while retaining the unrepaired state as a paired reference. A frozen SLM then evaluates the future solution domains induced by both states under a matched budget and determines whether to adopt the intervened reasoning result. On five mathematical reasoning benchmarks, SDTT improves the Qwen3-1.7B SLM's Math Avg. by 34.64% relative to greedy SLM decoding. This result indicates that effective small–large model collaboration depends not on optimizing the current token in isolation, but on whether the intervention changes the state on which the SLM relies for continued reasoning in a way that improves access to reliable solution domains within a finite budget.

1 Introduction

Token-level small–large model collaboration offers a fine-grained trade-off between reasoning capability and computational cost: the small language model (SLM) generates most reasoning steps, while the large language model (LLM) provides local assistance only at selected states (Huang et al., 2026; Zhao et al., 2026). This collaboration is particularly important for mathematical reasoning, because each local decision affects not only the current output but also the conditions on which subsequent derivation depends. The SLM may make a superficially plausible decision that it cannot reliably use to continue, such as producing an algebraic transformation that omits necessary intermediate steps. Even when the LLM identifies a disagreement at the same prefix, simply replacing the next token leaves the historical prefix and the preceding key-value (KV) state intact, and therefore may not alter the subsequent reasoning trajectory. Effective collaboration thus depends not only on when to invoke the LLM, but also on whether the local assistance truly improves the state on which the SLM's continued reasoning relies.

Some token-level collaborative decoders implement local assistance as next-token replacement or reranking. Select to Think (S2T) converts teacher generation into a choice among candidate tokens proposed by the student, while FutureBridge trains a reranker with answer-verified future trajectories and judges candidates by how effectively the student SLM can use them in subsequent reasoning. FusionRoute likewise combines expert selection with local logit correction (Ye et al., 2026; Li et al., 2026b; Xiong et al., 2026). These methods move beyond relying solely on the LLM's local preference by considering the effect of a candidate on the student's subsequent generation. However, their intervention still acts on

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.

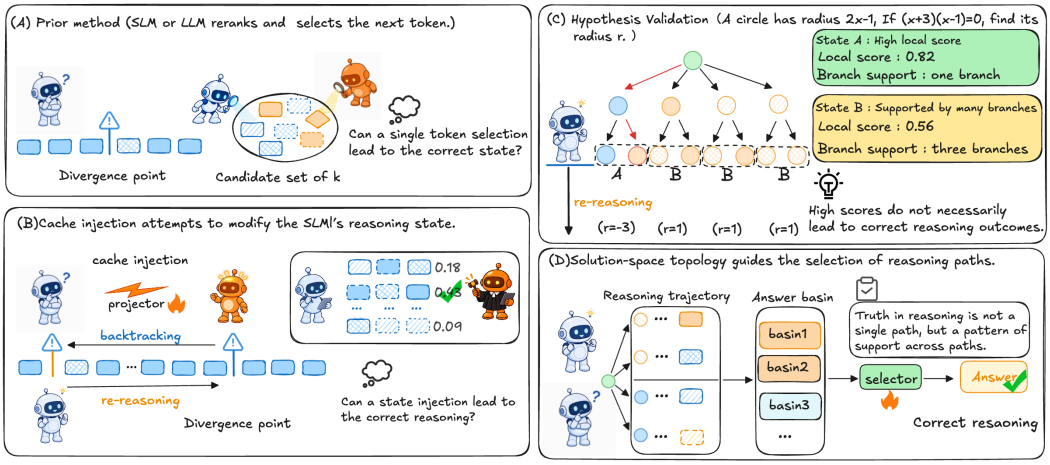

Figure 1: A collaborative reasoning framework for solution-domain topology selection. The figure contrasts token-level replacement, rollback-based state repair, and paired evaluation of future solution domains. The proposed method retains repaired and unrepaired states at the same rollback anchor, expands recovery trees under a matched SLM budget, groups terminal trajectories with identical predicted answers into answer basins, and makes the final selection using branch-lineage and candidate-structure information.

the current token while retaining the preceding prefix and its internal KV state. If the mismatch originates from an earlier reasoning decision or from how the SLM encodes the retained context, replacing or reranking the next token may still fail to eliminate the adverse state. The unresolved question is therefore not only which token to choose, but whether the state on which the SLM relies for continued reasoning should first be recovered. Figure 1 contrasts token-level replacement with cache-based state intervention and summarizes how paired future trees support solution-domain selection.

An intuitive way to repair the prior state is to roll back to an earlier prefix, undo some historical decisions, and modify the corresponding internal state. Prior work has separately explored influential-token intervention and cross-model cache transfer (Lin et al., 2024; Fu et al., 2026; Lee et al., 2026), but these techniques do not by themselves specify how to compare the future continuations induced by repaired and unrepaired states. Once the state is modified, the system faces another problem: the new state may give rise to multiple different subsequent rollouts. One approach is to evaluate one rollout directly and choose the trajectory that appears more reasonable or to be of higher quality. However, the local quality of a single rollout does not guarantee a correct final solution; a trajectory may be smoother at the current step while still leading the SLM down an incorrect reasoning path. State repair determines where to continue, but it does not automatically determine whether continuation can reach a correct solution. Subsequent selection therefore cannot rely only on the local score of a single rollout; it must also examine the overall structure of the futures reachable from the repaired state.

After state repair, the object of evaluation should not be limited to a single rollout, but should instead be the finite-budget solution domain induced by the state. We represent this domain with a fixed-budget set of sampled continuations and their branch structure, rather than claiming to enumerate all possible futures. Prior multi-continuation approaches such as Self-Consistency sample diverse reasoning paths and aggregate their answers, while Tree of Thoughts and Graph of Thoughts explicitly organize alternative paths and their relations; MARS^2 and BASIN use structural or allocation mechanisms during search (Wang et al., 2023; Yao et al., 2023; Besta et al., 2023; Li et al., 2026a; Cheng, 2026). However, these lines of work do not directly compare the future solution domains induced by repaired and unrepaired states under a matched budget after an intervention.

We therefore propose the Solution-Domain Topology Tree: the unrepaired and repaired states at the same rollback anchor serve as two roots, and a frozen SLM expands finite recovery trees from them under a matched budget. Terminal trajectories with the same

predicted answer are grouped into answer basins, and their branch lineages are retained. A local selector then compares candidates using basin structure and generation information. The original no-intervention output is retained as a fallback when the selected candidate fails a predefined threshold. This design unifies state repair, future-candidate reachability, and final selection within a single recovery framework, rather than treating the local quality of one rollout or answer frequency as a direct proxy for final correctness.

Our main contributions are as follows.

1. **We identify two gaps in token-level small–large model collaboration:** (i) local output correction cannot by itself undo the already committed prefix or KV state; and (ii) state repair alone does not necessarily lead to a correct candidate trajectory.
2. **We propose a disagreement-triggered paired state-repair mechanism:** when a disagreement is detected, it performs fixed rollback to a predefined anchor, uses a learnable cache-to-cache (C2C) projection to construct the repaired state, and preserves the unrepaired state at the same rollback anchor.
3. **We propose a lineage-aware solution-domain selection mechanism:** it expands candidate reasoning trees from the repaired and unrepaired states and selects among the resulting candidates using branch-lineage, basin-structure, and generation information.

2 Related Work

Token-level collaborative decoding. Token-level collaborative decoding studies when a stronger model should intervene during student generation, how much assistance it should provide, and how the receiving model should use that assistance. RelayLLM learned when the small model should initiate a handoff, while TrigReason, MentorCollab, and PyroDash used event or disagreement signals to trigger brief or one-shot interventions (Huang et al., 2026; Zhao et al., 2026; Wang et al., 2026; Lyu et al., 2026). For token-level output selection, FusionRoute combined expert selection with logit correction, Select to Think (S2T) turned teacher generation into candidate-token selection, and FutureBridge used answer-verified future trajectories during training to supervise the receiving student's compatibility (Xiong et al., 2026; Ye et al., 2026; Li et al., 2026b). However, replacing a token at a single disagreement point is not equivalent to repairing the state: when the original prefix and its KV representations are retained, the new token may fail to change the conditions on which the student's subsequent reasoning depends. A detected disagreement is only an observed behavioral outcome; it neither proves that the student is wrong nor identifies the earliest source of difficulty. Therefore, disagreement should be treated as a recovery signal, with the system first rolling back to an earlier prefix and repairing the corresponding internal state before selecting candidates and continuing reasoning.

Cross-Model Cache Communication. Cross-model cache communication provides mechanisms for changing the internal conditions of a receiving model. Cache-to-Cache (C2C) aligns, maps, and fuses the KV caches of source and receiving models, while Latent Space Communication via K-V Cache Alignment and Latent Cache Flow study shared latent interfaces and transformations of K/V representations (Fu et al., 2026; Dery et al., 2026; Rossi et al., 2026). Dense Latent Communication further distinguishes receiving settings that retain or discard the original context, while Mixture-of-Translators models target-side replay computation and conversion error (Chen et al., 2026; Lee et al., 2026). RelayCaching and Cross-Model KV Cache Transfer emphasize cache reuse and prefill acceleration within or across model families (Geng et al., 2026; Heo et al., 2026). These works clarify how representations can be transferred, reconstructed, or reused, but computational fidelity after conversion is not equivalent to a corrected reasoning state. Therefore, even after state repair, subsequent trajectories generated from different repaired states must still be organized, compared, and selected.

Structure-Aware Search. Model collaboration has also been combined with search, but existing methods typically intervene at different stages. AlphaRouter learns routing through collaborative reasoning-tree search; MOSA integrates multi-model proposals and revisions into Monte Carlo tree search; and LiteCoOp jointly selects program transformations and participating models in a compiler-optimization tree (Liao et al., 2026; Yang et al., 2026; Tang et al., 2026). MARS^2 trains search policies using parent–child and sibling relations, Local Branch Routing retains and prunes future states computed locally, and BASIN organizes exploration around intermediate reasoning strategies and repeated-visit penalties (Li et al., 2026a; Yin et al., 2026; Cheng, 2026). These methods show that branch relations and subsequent computation can guide search, but they do not define a finite, paired recovery tree grown by a frozen student from repaired and unrepaired roots. Our setting instead applies one disagreement-triggered rollback and cache intervention, lets the large model exit, and selects among answer basins while retaining root provenance and branch support. This lineage-preserving solution-domain structure is a finite selection representation rather than a correctness guarantee, so its value must be evaluated beyond answer counts and extra sampling.

3 Problem Statement

Given a mathematical problem x, let π_S and π_L denote the small and large models, respectively. At a reasoning disagreement state s_D, the system rolls back to an unrepaired root s_0 and uses local information from the large model to repair the small model's internal state, obtaining

$$s_1 = \rho_\phi(s_0; \pi_L), \tag{1}$$

where ρ_ϕ denotes the state-repair process parameterized by the projector parameters ϕ.

The small model then generates candidate trajectory sets from the corresponding states within a finite generation budget h. Let $\mathcal{R}$ denote the set of recovery roots and let K_h denote the number of candidates generated from each root:

$$\mathcal{C}_h(s_i) = \left\{ y_i^{(k)} \right\}_{k=1}^{K_h}, \qquad i \in \mathcal{R}. \tag{2}$$

The current implementation uses two recovery roots, corresponding to the unrepaired and repaired states. The system also retains the original no-intervention output y_{orig} and selects between the original output and the recovery candidates using their trajectories and structural features:

$$\hat{y} = g_\theta(y_{\text{orig}}, \{\mathcal{C}_h(s_i)\}_{i \in \mathcal{R}}), \tag{3}$$

where g_θ denotes the candidate-selection rule parameterized by θ. At deployment, g_θ cannot access the reference answer or direct evaluations of the candidates by the large model.

The objective is to maximize the probability of a correct final output under a finite computation budget B:

$$\max_{\phi,\theta} \ \Pr(\hat{y} = y^\star \mid x) \qquad \text{s.t.} \quad \text{Cost}(x; \phi, \theta) \le B, \tag{4}$$

where ϕ denotes the projector parameters, θ denotes the selector parameters, and $y^\star$ denotes the correct answer. The cost includes disagreement detection, teacher prefill and cache fusion, rollback, student-only tree expansion, and candidate selection.

4 Method

Our method has four stages. A frozen disagreement rule identifies an eligible boundary, and fixed rollback defines the recovery anchor. A learnable C2C map then constructs a

repaired SLM state while preserving the unrepaired state. The frozen SLM expands a paired solution-domain topology tree from both roots under a matched budget. Finally, a shared lineage-aware selector compares candidate basins and either returns a selected candidate or retains the original output. The projector and selector are trained offline; after cache injection, no further LLM calls are made.

4.1 Disagreement Localization and Fixed Rollback

We use the frozen S2T boundary rule $d_{\text{aud}}(t)$ to locate the first eligible disagreement. Let p_0 be the prompt length and let r_{rb} denote the rollback length. We evaluated $r_{\text{rb}} \in \{1, 3, 5, 7, 9\}$ on the development split and selected $r_{\text{rb}} = 7$; this value was then fixed for all reported evaluations. Let

$$\begin{aligned} \mathcal{T}_{\text{elig}} &= \{t : d_{\text{aud}}(t) = 1,\ t \geq p_0 + r_{\text{rb}}\}, \\ D &= \begin{cases} \min \mathcal{T}_{\text{elig}}, & \mathcal{T}_{\text{elig}} \neq \emptyset, \\ \perp, & \mathcal{T}_{\text{elig}} = \emptyset, \end{cases} \\ (A, s_0) &= \big(D - r_{\text{rb}}, R_A(s_D)\big) \quad \text{if } D \neq \perp. \end{aligned} \tag{2}$$

Here R_A restores the true prefix, cache, position, and mask at location A, revoking the generated state after that location. When $D \neq \perp$, Eq. (2) defines the unrepaired recovery root; when $D = \perp$, no recovery is triggered and the original small-model output is returned. The boundary rule is fixed by the implementation; its thresholds and divergence form are treated as implementation details and are not replaced here by an unverified formula.

Disagreement triggers recovery but is not treated as a causal error location. Fixed rollback separates anomaly localization from state intervention, while the development sweep determines the value used in the reported protocol. We also preserve the original generation state s_{orig} and its output; for stochastic generation, the random-number state or generated trajectory is saved to support exact recovery.

4.2 C2C State Repair

On the same retained prefix after rollback, C2C (Cache-to-Cache) maps and fuses the large-model cache into the small-model cache. Let K_L denote the large-model cache and $K_S^{(0)}$ the unrepaired cache, and use F_ϕ to denote the learnable projection and fusion module. The state transformation is defined by

$$K_S^{(1)} = F_\phi\Big(K_S^{(0)}, \text{Align}(K_L)\Big), \qquad s_1 = J_\phi(s_0, K_L). \tag{3}$$

Align aligns prefix positions, tokens, and layer/head interfaces, while J_ϕ denotes the state transformation after the cache update in Eq. (3). The projection is a learnable cross-model representation map; it need not be orthogonal or idempotent, and the two caches need not become numerically identical. Its purpose is to provide an internal condition that the SLM can use for continued reasoning, rather than to replace an isolated token. The map J_ϕ implements the state-repair operator ρ_ϕ defined in the problem statement.

C2C Projection Training Objective. Following the prefix-aligned training specification, we freeze both models and optimize only the projection/fusion parameters ϕ, using a correct continuation y^+ compatible with the retained prefix. Let s_A denote the small-model state at rollback anchor A before cache fusion:

$$\mathcal{L}_{\text{C2C}}(\phi) = -\mathbb{E}_{(s_A, y^+) \sim \mathcal{D}_{\text{fuse}}} \sum_{t=1}^{|y^+|} \log \pi_S\big(y_t^+ \mid J_\phi(s_A, K_L), y_{<t}^+\big)\,. \tag{4}$$

We train the projector offline with Eq. (4) while keeping both base models frozen, and freeze the learned projector before deployment. The supervised continuation is used only in this offline loss and cannot enter the large-model cache at deployment.

After constructing s_1, the LLM exits. All subsequent generation from both states is performed by the frozen SLM, and recovery-tree collection and candidate ranking make no further LLM calls.

4.3 Solution-Domain Topology Tree

The repaired and unrepaired states, s_1 and s_0, form paired roots. From each root, we expand a fixed $N_{\text{br}} \times N_{\text{cont}}$ recovery tree: N_{br} bridge branches are generated first, and each bridge produces N_{cont} subsequent trajectories. Let a leaf trajectory be ℓ_{cbr}, where $c \in \mathcal{R}$, $b \in \{1, \ldots, N_{\text{br}}\}$, and $r \in \{1, \ldots, N_{\text{cont}}\}$. The current implementation uses $N_{\text{br}} = 4$ and $N_{\text{cont}} = 2$ with two roots, giving sixteen logical leaf trajectories in total. The two conditions use the same completion policy and per-root budget. The expansion preserves shared prefixes and branch lineages and performs no adaptive pruning or value backup.

We use answer basins rather than individual trajectories as the selection unit. A frozen parser extracts the predicted answer $\hat{a}(\ell)$. On the valid-leaf set $\mathcal{L}_c^{\text{valid}}$ of each tree, define the answer equivalence relation

$$\ell \sim_\nu \ell' \iff \nu(\hat{a}(\ell)) = \nu(\hat{a}(\ell')), \qquad \mathcal{B}_c = \mathcal{L}_c^{\text{valid}} / \sim_\nu . \tag{5}$$

Each equivalence class B_{cj} in $\mathcal{B}_c$ is an answer basin. The label-free normalization function ν reads only the student prediction and not the reference label; invalid leaves retain a failure marker and are not merged into a valid basin. Grouping applies only to terminal leaves: intermediate cache states are not merged, and equal answers do not imply equivalent proof semantics.

Answer equivalence alone loses branch provenance, so we also retain the bridge–basin incidence matrix

$$W_{cbj} = \sum_{r=1}^{N_{\text{cont}}} \mathbf{1}\{\ell_{cbr} \in B_{cj}\}. \tag{6}$$

The basin representation combines terminal answer groups with bridge incidence, distinguishing repeated answers within one branch from support shared by multiple branches. Basin mass and cross-branch coverage are

$$m_{cj} = \frac{|B_{cj}|}{N_{\text{br}} N_{\text{cont}}}, \qquad \kappa_{cj} = \frac{|\{b : W_{cbj} > 0\}|}{N_{\text{br}}}. \tag{7}$$

For each basin we select a label-free medoid trajectory and extract branch coverage, path association, answer concentration, and generation-probability features for ranking. Equation (7) gives two of these features; the complete d_x-dimensional feature set, raw entropy and structure scores, and representative-selection rule are given in Appendix D. In our implementation, $d_x = 18$. This is a finite-sample representation of answer grouping and lineage statistics; its stability can inform selection but does not guarantee correctness.

4.4 Lineage-Aware Candidate Selection

Across the two root conditions, a shared lightweight selector compares candidate basins generated for the same input state. Its parameters are trained offline across states, while deployment uses only features from the paired trees and does not access the reference answer or call the LLM.

For candidate $i = (c, j)$, we extract a d_x-dimensional feature vector x_i without correctness labels, covering basin structure, generation probability, completion, and budget information.

After standardization with training-subset statistics, it is fed to a $d_x \to h_1 \to h_2 \to 1$ multilayer perceptron defined in Eq. (8). Our implementation uses $d_x = 18$, hidden widths $h_1 = 32$ and $h_2 = 16$:

$$\tilde{x}_{ik} = \frac{x_{ik} - \mu_k}{\max(\sigma_k, \varepsilon_{\text{std}})}, \qquad f_\theta(x_i) = W_3 \,\text{ReLU}(W_2 \,\text{ReLU}(W_1 \tilde{x}_i + b_1) + b_2) + b_3. \tag{8}$$

The standardization floor is $\varepsilon_{\text{std}} = 10^{-6}$ in our implementation. The resulting ranker has 1,153 parameters and makes decisions at the candidate-basin level rather than for isolated leaves. Offline training constructs correct–incorrect candidate pairs $\mathcal{P}_s$ within each state s and uses the state-normalized pairwise logistic loss in Eq. (9) (Burges et al., 2005):

$$\mathcal{L}_{\text{pair}}(\theta) = \frac{1}{|\mathcal{S}_\pm|} \sum_{s \in \mathcal{S}_\pm} \frac{1}{|\mathcal{P}_s|} \sum_{(i^+, i^-) \in \mathcal{P}_s} \log\Big(1 + e^{f_\theta(x_{i^-}) - f_\theta(x_{i^+})}\Big). \tag{9}$$

$\mathcal{S}_\pm$ contains training states with at least one correct–incorrect candidate pair. Other states do not contribute to this loss but remain in evaluation. State normalization prevents states with many candidates from dominating training merely through pair count. Correctness is used only as an offline pair label and cannot enter basin construction or online features.

At deployment, let $i^* = \arg\max_i f_\theta(x_i)$. A threshold selected on the development set and then frozen controls the final output according to Eq. (10):

$$\hat{y} = \begin{cases} \text{Medoid}(B_{i^*}), & f_\theta(x_{i^*}) \geq \tau_{\text{dev}}, \\ y_{\text{orig}}, & \text{otherwise.} \end{cases} \tag{10}$$

When no valid candidate exists, the saved original output y_{orig} is returned as well. The model, standardization statistics, and threshold are jointly frozen, and independent-holdout labels do not participate in selection. The raw ranking score is not interpreted as a success probability or repair gain; analysis of relative gain, selection error, and gating conditions is in Appendix B.2, and the complete training and deployment procedure is in Appendix A.

5 Experiments

This section is organized around three questions: whether cache intervention adds value beyond extra student sampling, whether repair changes the reachable future solution domain, and whether lineage-aware selection improves the final decision. The evaluation reports end-to-end accuracy, inference cost, mechanism ablations, candidate diagnostics, and robustness factors under the protocol defined below.

5.1 Experimental Setup

Benchmarks and metrics. The protocol evaluates GSM8K, MATH-500, OlympiadBench, AIME 2024, and AIME 2025. The primary metric is single-trajectory pass@1 accuracy, and Math Avg. is the unweighted mean over the five benchmarks. All methods use the same zero-shot chain-of-thought prompt and answer verifier, with generation truncated after 4,096 output tokens.

Models and baselines. The primary setting uses Qwen3-1.7B as the SLM and Qwen3-32B as the LLM; Qwen3-0.6B is used for scale generalization. Base model parameters remain frozen. The core comparisons are greedy SLM decoding, token-level baselines including FutureBridge (Li et al., 2026b), matched no-op search, fixed C2C repair, and the full Solution-Domain Topology Tree.

Budget protocol. The no-op and repaired conditions use the same student rollout budget. The no-op+no-op control matches the two-root sampling budget without cache intervention. Post-injection tree collection does not call the LLM again.

Table 1: End-to-end accuracy (%) on five mathematical reasoning benchmarks. FutureBridge is included as a direct token-selection baseline.

Param.	Benchmark	SLM-only			LLM-involved				
		Greedy	Maj@8	S2T-Local	R2R	Takeover	S2T	FutureBridge	SDTT
1.7B	GSM8K	82.0	83.5	86.7	95.3	95.5	95.8	96.0	96.2
	MATH-500	64.0	65.0	67.6	78.4	79.0	79.7	80.6	80.4
	OlympiadBench	27.1	27.9	28.9	41.5	42.0	42.4	43.1	44.5
	AIME 2024	5.6	6.7	7.8	13.3	14.4	15.6	15.6	15.6
	AIME 2025	7.8	8.9	10.0	10.0	11.1	12.2	12.2	14.4
	Math Avg.	37.30	38.40	40.20	47.70	48.40	49.14	49.50	50.22
0.6B	GSM8K	57.8	66.2	68.5	76.0	76.8	77.5	76.8	79.2
	MATH-500	33.5	41.5	47.2	52.5	53.0	54.2	53.2	55.0
	OlympiadBench	13.2	18.0	20.5	27.0	28.0	28.5	26.7	29.0
	AIME 2024	4.4	6.7	7.8	10.0	11.1	12.2	10.0	13.3
	AIME 2025	3.3	5.6	6.7	7.8	8.9	10.0	7.8	11.1
	Math Avg.	22.44	27.60	30.14	34.66	35.56	36.48	34.90	37.52

Table 2: Token-level inference cost on GSM8K for the primary Qwen3-1.7B SLM. Accuracy is problem-level pass@1; calls are averaged per problem and latency is normalized to greedy SLM decoding. FutureBridge is included for direct accuracy comparison; its cost trace is not reported in this table.

Method	Accuracy	Calls	Latency
Greedy SLM	82.0	0.0	1.00×
S2T	95.8	4.8	3.18×
FutureBridge	96.0	4.8	3.80×
SDTT	96.2	5.3	4.83×

The GSM8K row in Table 1 is the primary benchmark anchor for the Qwen3-1.7B setting. The budget, mechanism-ablation, and candidate-diagnostic tables below use the same GSM8K evaluation protocol; the other rows of Table 1 remain separate benchmark results.

For rollback selection, we evaluated $r_{\text{rb}} \in \{1, 3, 5, 7, 9\}$ on the development split and fixed $r_{\text{rb}} = 7$ for all reported evaluations.

5.2 Main Results

End-to-end accuracy. Table 1 follows the established benchmark layout for token-level collaborative decoding. The final method column reports the Solution-Domain Topology Tree (SDTT).

The main comparison should be interpreted together with the matched-budget token-level comparison below. A gain over greedy decoding alone does not distinguish the effect of the collaborative method from the baseline policy.

5.3 Mechanism and Budget Ablations

Accuracy under a matched collaboration budget. The table isolates token-level collaboration cost for the primary Qwen3-1.7B setting and reports accuracy, teacher-call frequency, and normalized latency.

State-repair and selector ablation. The following ablation separates the effect of cache intervention, paired roots, and lineage-aware selection. The values report the latest five-condition ablation run.

The Paired C2C condition is included as a separate paired-root control; all five rows use the same evaluation protocol. The SDTT accuracy entry is the GSM8K value reported in the primary benchmark table, so the main result and the mechanism ablation use the same accuracy definition.

Table 3: State-repair and solution-domain selection ablations on GSM8K. Candidate reachability and final accuracy are percentages; final accuracy is problem-level pass@1.

Variant	State repair	Paired roots	Lineage features	Selector	Candidate reachability	Accuracy
No-op16	No	No	No	Fixed	94.87	95.10
No-op+No-op	No	Yes	No	Fixed	95.42	95.28
Fixed C2C	Yes	No	No	Fixed	94.16	95.71
Paired C2C	Yes	Yes	No	Basin mass	94.11	95.12
SDTT	Yes	Yes	Yes	Lineage-aware	94.63	96.20

Table 4: Candidate reachability and final-selection diagnostics on GSM8K. Correct candidate is the fraction of states with at least one correct leaf; correct-basin mass is the mean leaf mass assigned to correct answer basins. All entries are percentages; W2C and C2W are transition rates and Fallback is the fallback rate.

Condition	Correct candidate	Correct basin mass	Selected accuracy	W2C	C2W	Fallback
No-op16	94.87	98.45	94.72	4.12	1.23	5.32
No-op+No-op	95.42	93.87	92.94	4.26	1.86	5.68
Fixed C2C	94.16	93.52	93.28	3.87	1.34	6.21
SDTT	94.63	93.16	96.20	4.18	1.95	6.43

5.4 Candidate Reachability and Final Selection

Candidate reachability and final selection must be reported separately. A correct candidate can be present without being selected, while a final gain can otherwise be attributed incorrectly to the selector. The paired metrics in Table 4 are therefore required for the main claim.

All four rows in this table come from the latest reported ablation run. The SDTT selected-accuracy entry is the verified GSM8K value and is aligned with the final-accuracy value above. Under these entries, SDTT is the highest final-accuracy condition; the other diagnostic columns measure candidate coverage, basin structure, and intervention transitions rather than final accuracy and therefore need not be monotone across variants.

5.5 Robustness and Failure Analysis

The robustness protocol and failure taxonomy are specified in Appendix C.4.

6 Conclusion

We have recast small–large model collaboration as a post-injection recovery problem under a finite budget. We have proposed the Solution-Domain Topology Tree, which pairs repaired and unrepaired states at the same rollback anchor, expands frozen-SLM recovery trees under a matched budget, and selects candidates using answer basins and branch structure. This framework unifies local state intervention with future solution-domain evaluation and provides a structured basis for studying accuracy gains in collaborative decoding.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023. URL https://arxiv.org/abs/2308.09687.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pp. 89–96, 2005. doi: 10.1145/1102351.1102363. URL https://www.microsoft.com/en-us/research/publication/learning-to-rank-using-gradient-descent/.

Siyi Chen, Xiaoyan Zhang, Meng Wu, Jonathan Tremblay, Valts Blukis, Stan Birchfield, Rene Vidal, Alvaro Velasquez, Sijia Liu, and Qing Qu. See what i see, know what i think: Dense latent communication across heterogeneous agents. *arXiv preprint arXiv:2606.13594*, 2026. URL https://arxiv.org/abs/2606.13594.

Lu Cheng. Escaping redundant reasoning: Structure-aware search for inference-time LLMs. *arXiv preprint arXiv:2609.00738*, 2026. URL https://arxiv.org/abs/2609.00738.

Lucio M. Dery, Zohar Yahav, Henry Prior, Qixuan Feng, Jiajun Shen, and Arthur Szlam. Latent space communication via K-V cache alignment. *arXiv preprint arXiv:2601.06123*, 2026. URL https://arxiv.org/abs/2601.06123.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. In *International Conference on Learning Representations (ICLR 2026)*, 2026. URL https://arxiv.org/abs/2510.03215.

Yingsheng Geng, Yuchong Gao, Weihong Wu, Guyue Liu, and Jiang Liu. Relay-caching: Accelerating LLM collaboration via decoding KV cache reuse. *arXiv preprint arXiv:2603.13289*, 2026. URL https://arxiv.org/abs/2603.13289.

Taekyung Heo, Rasoul Shafipour, Ritchie Zhao, Maximilian Golub, Mohammad Mahdi Kamani, Ritika Borkar, Makesh Tarun Chandran, Pantea Zardoshti, and Bita Darvish Rouhani. Cross-model KV cache transfer in LLM families: A closed-form linear mapping for prefill reuse. *arXiv preprint arXiv:2608.03893*, 2026. URL https://arxiv.org/abs/2608.03893.

Chengsong Huang, Tong Zheng, Langlin Huang, Jinyuan Li, Haolin Liu, and Jiaxin Huang. RelayLLM: Efficient reasoning via collaborative decoding. *arXiv preprint arXiv:2601.05167*, 2026. URL https://arxiv.org/abs/2601.05167.

Jin-woo Lee, Minkyung Song, Junghyun Oh, Seunghoon Han, Soyoung Park, Gwangseon Jang, and Sungsu Lim. Mixture-of-translators: Translating KV caches across heterogeneous large language models. *arXiv preprint arXiv:2607.28979*, 2026. URL https://arxiv.org/abs/2607.28979.

Pengfei Li, Shijie Wang, Fangyuan Li, Yikun Fu, Kaifeng Liu, Kaiyan Zhang, Dazhi Zhang, Yuqiang Li, Biqing Qi, and Bowen Zhou. MARS^2: Scaling multi-agent tree search via reinforcement learning for code generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 33319–33335, 2026a. doi: 10.18653/v1/2026.acl-long.1538. URL https://aclanthology.org/2026.acl-long.1538/.

Quanquan Li, Hongbo Zhang, Yihe Chi, Jingyu Li, Xidong Xi, Liuyang Song, Hongzhen Zhang, Yuxiang Huang, Jing Ke, Siyuan Ma, Junyi Lin, and Guitao Cao. FutureBridge: Token selection beyond local preference in collaborative decoding. *arXiv preprint arXiv:2608.06819*, 2026b. URL https://arxiv.org/abs/2608.06819.

Siteng Liao, Yuzhu Liang, Hengzhong Rao, Xizhao Luo, and Tian Wang. Alpharouter: Token-level routing between SLM and LLM with reinforcement learning and tree search. In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, 2026. URL https://openreview.net/forum?id=TWxFLz7xen.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances LLM's reasoning capability. *arXiv preprint arXiv:2411.19943*, 2024. URL https://arxiv.org/abs/2411.19943.

Niqi Lyu, Pengtao Shi, Wei Qiu, Jianlin Zhong, Sicong Xia, Jianyao Ma, and Yicheng Ding. Pyrodash: Cost-efficient token-level small-large language model collaborative inference. *arXiv preprint arXiv:2607.20327*, 2026. URL https://arxiv.org/abs/2607.20327.

Maximillian Rossi, Prajwal Raghunath, and Eugene Wu. Latent cache flow: Model-to-model communication without text. *arXiv preprint arXiv:2605.22863v2*, 2026. URL https://arxiv.org/abs/2605.22863v2.

Annabelle Sujun Tang, Christopher Priebe, Lianhui Qin, and Hadi Esmaeilzadeh. LiteCoOp: Lightweight multi-LLM shared-tree reasoning for model-serving compiler optimizations. *arXiv preprint arXiv:2602.01935v2*, 2026. URL https://arxiv.org/abs/2602.01935v2.

Haojin Wang, Yike Wang, Shangbin Feng, Hannaneh Hajishirzi, and Yulia Tsvetkov. Mentor-collab: Large-to-small inference-time mentorship for concise reasoning in language models. *arXiv preprint arXiv:2602.05307v3*, 2026. URL https://arxiv.org/abs/2602.05307v3.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR 2023)*, 2023. URL https://arxiv.org/abs/2203.11171.

Nuoya Xiong, Yuhang Zhou, Hanqing Zeng, Zhaorun Chen, Furong Huang, Shuchao Bi, Lizhu Zhang, and Zhuokai Zhao. Token-level LLM collaboration via fusionroute. In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, 2026. URL https://arxiv.org/abs/2601.05106v5.

Sen Yang, Yafu Li, Wai Lam, and Yu Cheng. Multi-LLM collaborative search for complex problem solving. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42599–42614, 2026. doi: 10.18653/v1/2026.findings-acl.2115. URL https://aclanthology.org/2026.findings-acl.2115/.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. URL https://arxiv.org/abs/2305.10601.

Wenxuan Ye, Yangyang Zhang, Xueli An, Georg Carle, and Yunpu Ma. Select to think: Unlocking SLM potential with local sufficiency. *arXiv preprint arXiv:2604.26940*, 2026. URL https://arxiv.org/abs/2604.26940.

Yutong Yin, Mingyu Jin, Jin Pan, Changyi Yang, Zijie Xia, Dhruv Pai, Shuming Hu, Zhen Zhang, Chenyang Zhao, Jinman Zhao, Wujiang Xu, Raymond Li, Xin Eric Wang, Julian McAuley, and Zhaoran Wang. Efficient and trainable language model test-time scaling via local branch routing. *arXiv preprint arXiv:2606.25354v2*, 2026. URL https://arxiv.org/abs/2606.25354v2.

Yi Zhao, Yajuan Peng, Cam-Tu Nguyen, Zuchao Li, Xiaoliang Wang, Xiaoming Fu, and Hai Zhao. Trigreason: Trigger-based collaboration between small and large reasoning models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 6700–6718, 2026. doi: 10.18653/v1/2026.findings-acl.333. URL https://aclanthology.org/2026.findings-acl.333/.

A Implementation Details and Deployment

This appendix records the implementation details needed to reproduce the reported protocol. It distinguishes the frozen mainline from exploratory audits and does not include unimplemented graph or committor variants.

A.1 Models, data splits, and rollback selection

The primary configuration uses Qwen3-1.7B as the SLM and Qwen3-32B as the LLM; Qwen3-0.6B is used for the smaller-model setting. Both base models are frozen during deployment. The S2T disagreement rule is frozen before evaluation. We evaluated rollback lengths $r_{\mathrm{rb}} \in \{1, 3, 5, 7, 9\}$ on the development split and selected $r_{\mathrm{rb}} = 7$. This value is fixed for all reported test and mechanism runs; no test labels are used in this choice.

The frozen $r7$ projector collection contains 8,237 raw repair-state records, 8,110 records after quality filtering, and 7,926 complete paired records. The later mechanism audit uses a separate exploratory split of 124 question groups, partitioned into four shards of 31 groups. References are stored in a separate audit manifest and are not exposed during generation or selection.

The main-text benchmark tables take priority for reported performance. In particular, the primary Qwen3-1.7B GSM8K result for SDTT is 96.2% in Table 1; the budget, mechanism-ablation, and candidate-diagnostic tables in the main text use the same GSM8K evaluation protocol. The supplementary audits below use explicitly different splits or question-group manifests and therefore are not alternative estimates of that benchmark result.

A.2 C2C projector training

The projector is trained offline with the loss in Eq. (4) using frozen Qwen3-1.7B and Qwen3-32B models. The three-layer projector uses width 1,024, dropout 0.1, zero initialization, and a temperature schedule from 1.0 to 0.001 over 2,000 updates. Training uses one epoch, learning rate 10^{-4}, weight decay 0.01, gradient accumulation 8, gradient clipping 1.0, contrastive weight 0.1, contrastive margin 0.2, and random seed 42. The projector is frozen before mechanism evaluation. Reference answers are used only for offline filtering and audit; they are absent from the deployed cache-injection and tree-selection operations.

A.3 Deployment procedure

The deployed procedure is:

1. Generate the original SLM path and save its recoverable cursor and output.
2. Find the first eligible disagreement D; if none exists, return the original output.
3. Set $A = D - 7$ and restore the prefix, KV cache, positions, and masks at A.
4. Construct the unrepaired root s_0 and the repaired root s_1 with one aligned C2C injection.
5. Release the LLM. From each root, generate four bridge branches and two continuations per branch.
6. Group terminal predictions into answer basins, compute the frozen feature vector, and select a medoid for each basin.
7. Apply the frozen selector across both roots; return the selected medoid only when its score exceeds the development threshold, otherwise return the saved original output.

Post-injection tree collection uses only the frozen SLM and makes no further LLM calls. The full cost includes disagreement detection, teacher prefill and fusion, rollback, student-tree expansion, and candidate selection; zero teacher calls during cached collection is therefore not an end-to-end teacher-free claim.

B Mathematical Analysis

The following results characterize finite-budget recovery and selection. They are sufficient conditions and error decompositions, not guarantees that every checkpoint satisfies their assumptions.

B.1 Finite-horizon reachability and repair

Let $\mathcal{S}$ include the prefix, cache, position, mask, and remaining budget. With correct terminals absorbed in $\mathcal{C}$ and incorrect or invalid terminals absorbed in $\mathcal{F}$, define

$$q_0(s) = \mathbf{1}\{s \in \mathcal{C}\}, \qquad q_h(s) = \int q_{h-1}(z) P_S(s, dz). \tag{5}$$

where P_S is the frozen SLM transition kernel. For paired roots, the repair effect at horizon h is $\Delta_h = q_h(s_1) - q_h(s_0)$. This definition counts over-budget or invalid completions as failures and does not assume a stationary distribution.

Proposition 1 (probability-mass transfer). Let $\nu_c = P_S(s_c, \cdot)$ be the successor distribution from root s_c. Then

$$\Delta_h = \int q_{h-1}(z)[\nu_1 - \nu_0](dz). \tag{6}$$

If $\nu_1 - \nu_0 = \epsilon(\mu_+ - \mu_-)$, $\epsilon > 0$, and $q_{h-1} \geq b$ on the support of μ_+ while $q_{h-1} \leq a$ on the support of μ_- with $b > a$, then $\Delta_h \geq \epsilon(b - a) > 0$.

Proof. Substitute the assumed measure decomposition into the preceding identity and bound the two integrals by b and a, respectively. The result requires complete cache states in the state representation; merging states by text prefix alone is insufficient. □

This proposition formalizes the intended role of C2C: it must shift future probability mass toward states that are easier for the SLM to complete, rather than merely changing the local token preference.

B.2 Selection error and relative gain

Let E_s denote that at least one correct candidate or the original fallback exists for state s. For any selector π,

$$\Pr(Y_\pi = 0) = \Pr(E_s^c) + \Pr(E_s, Y_\pi = 0). \tag{7}$$

The first term is a reachability failure and cannot be repaired by a ranker; the second is selection error. Relative to the original output,

$$\mathbb{E}[Y_\pi - Y_0] = \Pr(Y_0 = 0, Y_\pi = 1) - \Pr(Y_0 = 1, Y_\pi = 0), \tag{8}$$

which is the W2C rate minus the C2W rate.

Proposition 2 (complementarity beyond extra no-op sampling). Let E_0 and E_1 denote correct-candidate existence in the no-op and repaired trees, and let E_0' denote success of a second no-op tree with the same additional budget. Then

$$\Pr(E_0 \cup E_1) - \Pr(E_0 \cup E_0') = \Pr(E_0^c)\left[\Pr(E_1 \mid E_0^c) - \Pr(E_0' \mid E_0^c)\right]. \tag{9}$$

Proof. Expand each union as $\Pr(E_0) + \Pr(E_0^c \cap E)$ and subtract. □

The identity shows why the no-op+no-op control is necessary: a repaired tree is useful only when it supplies additional correct candidates on states where the first no-op tree fails.

B.3 Correlated tree sampling

Suppose B bridge states are independent and each bridge has R conditionally independent continuations with success probability Q_b. Then

$$P_{\text{hit}} = 1 - \left[\mathbb{E}(1 - Q)^R\right]^B. \tag{10}$$

For a fixed answer class with conditional probability $p(U)$, the leaf-fraction variance is

$$\text{Var}(\hat{m}) = \frac{\text{Var}(p(U))}{B} + \frac{\mathbb{E}[p(U)(1 - p(U))]}{BR}. \tag{11}$$

Increasing siblings reduces only the second term; increasing independent bridges reduces the shared-prefix term. Thus the 4×2 tree must not be interpreted as eight independent observations.

B.4 Limits of label-free structure

Mass, coverage, entropy, and lineage features cannot provide a universal correctness certificate without assumptions on the conditional relation between these observables and the answer. Two task distributions can have identical label-free trees while assigning different answers to the same terminal basins. The selector is therefore evaluated empirically against paired baselines rather than interpreted as a proof of correctness.

Table 5: Solution-Domain Topology Tree robustness protocol. The rollback sweep on the development split selects $r_{\mathrm{rb}} = 7$; the remaining rows record the fixed settings used in the reported evaluation.

Factor	Variation	Reported setting	Protocol
Rollback length	$r_{\mathrm{rb}} \in \{1, 3, 5, 7, 9\}$	$r_{\mathrm{rb}} = 7$	Development selection
Tree budget	Four bridges and two continuations	4×2 per root	Matched across roots
Ranker capacity	18–32–16 MLP	Frozen selector	Development threshold
Randomness	Fixed deployment seed	Seed 42	Same seed across conditions
Data isolation	Nested development and holdout	Separate manifests	No test-label access

Table 6: Frozen-projector holdout audit on 3,857 GSM8K examples.

Condition	Correct	Total	Accuracy	Wilson 95% CI
$r5$ matched reference	3,426	3,857	88.83%	[87.79,89.78]
$r7$ frozen projector	3,423	3,857	88.75%	[87.71,89.71]

C Supplementary Experiments

C.1 Robustness protocol

The robustness protocol varies rollback length, tree budget, selector capacity, random seed, and data isolation under the same question-group split and holdout convention used by the main evaluation.

C.2 Frozen-projector holdout audit

Before the topology mechanism study, the frozen projector was evaluated on a held-out remainder of 3,857 GSM8K examples with audited zero overlap with its calibration, collection, and training sources. The $r5$ condition is the matched reference and $r7$ uses the frozen rollback-7 projector. This is a projector holdout audit, not the primary end-to-end benchmark reported in Table 1; its percentages must not be substituted for the main-text GSM8K accuracy.

At the question level, 68 examples were corrected only by $r7$ and 71 were correct only under the $r5$ reference, giving a paired net change of -3. This audit characterizes the frozen projector and does not establish a gain for the later topology selector.

C.3 Exploratory H20 mechanism audit

The mechanism audit uses 124 question groups, four shards, and a 4×2 student tree from each root. Each rollout is limited to 1,024 new tokens. The LLM constructs the repaired root once and then exits; tree generation and selection use zero teacher forward calls, autoregressive teacher tokens, and teacher leaf-judge calls. The run uses physical GPUs 1–4 and is exploratory, not a replacement for the independent GSM8K benchmark evaluation or the main-text SDTT result.

Relative to the fixed f_7 reference, the repaired topology policy has 13 wins and 16 losses (paired net -3, McNemar $p = 0.7111$). Relative to repaired maximum-log-probability, it has 18 wins and 5 losses (net $+13$, $p = 0.01062$); relative to no-op topology, it has 8 wins and 14 losses (net -6, $p = 0.2863$). The topology score has audit AUC 0.8576 and 15 cases

Table 7: Exploratory H20 mechanism audit on 124 question groups.

Policy	Correct	Total	Accuracy
Fixed f_7 reference	72	124	58.06%
No-op topology policy	75	124	60.48%
Repaired topology policy	69	124	55.65%
Repaired maximum-log-probability	56	124	45.16%
Repaired eight-leaf oracle	95	124	76.61%

with a wrong dominant consensus. These values are retained as mechanism diagnostics, not as confirmatory benchmark claims.

C.4 Rollback and failure analysis

The rollback sweep uses the development split only and evaluates $r_{\mathrm{rb}} \in \{1, 3, 5, 7, 9\}$; $r_{\mathrm{rb}} = 7$ is then frozen. Failure records distinguish missing correct candidates, correct candidates not selected, wrong high-coverage basins, C2C damage, rollback context loss, parser errors, budget noncompletion, and correct answers with inconsistent reasoning. The generation manifest contains no reference answers; references enter only through the separate audit manifest after decoding.

D Feature Dictionary and Reproducibility

The selector uses 18 label-free features computed from student predictions; correctness labels and reference answers are unavailable online.

Table 8: Feature dictionary used by the frozen selector.

No.	Feature	Definition
1	Condition	Intervention indicator $\mathbf{1}\{c = 1\}$.
2	Basin mass	$\lvert B_j\rvert/(N_{\mathrm{br}} N_{\mathrm{cont}})$.
3	Bridge coverage	Fraction of bridges supporting basin B_j.
4	Sibling consistency	Fraction of sibling pairs assigned to the same basin.
5	Normalized LCP	Mean longest-common-prefix length, normalized by continuation length.
6	Path diversity	Within-basin path-diversity statistic.
7	Basin log probability	Mean token log probability of basin leaves.
8	Valid-leaf fraction	$V/(N_{\mathrm{br}} N_{\mathrm{cont}})$.
9	Valid-basin count	Number of valid answer basins normalized by leaf count.
10	Answer entropy	$-\sum_j (\lvert B_j\rvert/V) \log(\lvert B_j\rvert/V)$ for $V > 0$.
11	Same-answer sibling bridges	Number of bridge pairs supporting the same answer basin.
12	Structure score	$m_{\mathrm{dom}} c_{\mathrm{dom}} (V/L)(1 - H)$, with $L = N_{\mathrm{br}} N_{\mathrm{cont}}$.
13	Medoid bridge completion	Completion indicator for the selected medoid bridge.
14	Medoid greedy descendant	Indicator of greedy descendant generation.
15	Bridge log probability	Mean log probability over bridge tokens.
16	Continuation log probability	Mean log probability over continuation tokens.
17	Completion audit	Completion-status indicator.
18	Response length	Generated response tokens divided by the available budget.

The executed operations are rollback R_A, cache transformation J_ϕ, fixed 4×2 expansion Expand, answer projection Q_ν, incidence aggregation W, and label-free selection Select_θ. Basin representatives are medoid trajectories; intermediate states are not merged, and no persistent cross-question graph is constructed. Each run records model and tokenizer versions, trigger and rollback settings, tree dimensions, random seeds, shard assignments, parser version, threshold provenance, input and output hashes, and separate teacher-prefill and post-injection student costs. The selector, standardization statistics, projector, and development threshold are frozen before holdout evaluation.