Disagreement-Guided Test-Time Scaling for Structured Reasoning

Anonymous ACL submission

Abstract

Test-time scaling improves reasoning by generating more tokens, sampling more candidates, or searching more broadly. These methods share a structural limitation: they treat the reasoning trace as monolithic and cannot localize *where* within a chain the information deficit lies, so much of the added compute is spent on branches that contribute no new information. We propose a more diagnostic signal. At any reasoning step, a model maintains three coupled descriptions of the problem state—a surface explanation, a structural intermediate, and an outcome distribution—and the divergence among them is an information-theoretic lower bound on the unresolved uncertainty that further generation could in principle reduce. Building on this observation, we introduce Disagreement-Guided Test-Time Scaling (DTS), which organizes the reasoning state into a belief lattice over the three levels and allocates compute only at steps where cross-level disagreement is both detected and classified as actionable. A targeted branching procedure expands only the subproblem responsible for the conflict, and an amortized verifier controls whether further budget is warranted before full traces are generated. On structured reasoning benchmarks spanning mathematics, code, and long-chain inference, DTS attains higher accuracy under fixed compute while using 21.8% fewer tokens and 17.4% fewer branches than the strongest baselines, demonstrating that *localizing* the information deficit is a stronger scaling signal than token length or global uncertainty. The code is available in `https://anonymous.4open.science/status/DTTS-B6E2`.

1 Introduction

Reasoning with large language models increasingly depends on what happens at test time. Longer chains of thought, broader sampling, tree search, and verifier-guided reranking each show that additional compute improves accuracy (Wei et al., 2022; Yao et al., 2023; Wang et al., 2023). The improvement is uneven: some problems benefit substantially, while others consume budget without resolving the core uncertainty. The asymmetry has a structural cause.

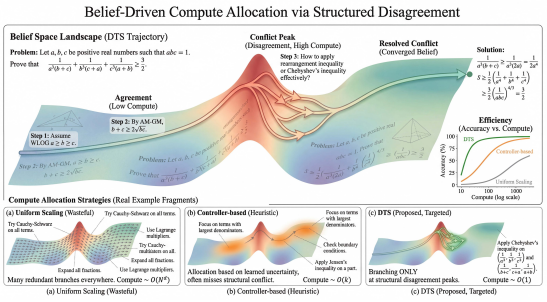

Figure 1: Uniform test-time expansion versus disagreement-guided allocation. Uniform allocation spreads compute evenly across reasoning steps. DTS detects cross-abstraction disagreement and concentrates additional budget on the subproblem where the reasoning state remains unresolved.

Best-of-N sampling, self-consistency, long chain-of-thought generation, tree search, and terminal verifier reranking all operate on the reasoning trace at a coarse granularity. They can detect that something is wrong—through low confidence, answer disagreement, or verifier rejection—but cannot localize *where* the information deficit lies. Their recourse is to expand the trace globally: more tokens, more candidates, more branches. The result is wasted compute on steps that are already informationally closed, and insufficient compute on the steps that actually carry the conflict (Figure 1).

We argue that a more diagnostic signal is intrinsic to the reasoning process itself. At any step, a model simultaneously maintains three descriptions of the same problem state: a surface explanation in natural language, a structural intermediate such as an equation template or program fragment, and an outcome distribution over candidate answers. When the three agree, the state is informationally

closed and further generation produces textual elaboration without informational gain. When they diverge, the model has evidence—internal to its own forward pass—that its description of the problem has not yet converged. The divergence is, as we show in §3.2, a lower bound on the residual entropy that further compute could in principle reduce.

This shifts test-time scaling from a question of trace expansion to one of information-directed intervention. The question is no longer "should the model think longer?" but "does the current state contain an unresolved conflict, and which subproblem is responsible?" Verification then changes role: instead of a terminal judge that selects among completed traces, the verifier becomes an early allocator that estimates whether a detected disagreement is worth resolving. Budget control operates *within* the reasoning loop rather than after it.

The proposed framework, Disagreement-Guided Test-Time Scaling, formalizes this view through three coupled components: a belief lattice that represents the state across surface, structural, and outcome levels and supports a partial order over informational closure; a disagreement detector that classifies divergence into wording variation, computation error, information insufficiency, or structural contradiction, with only the latter two triggering compute; and an amortized verifier that gates targeted branching by expected marginal information gain. The same compute that uniform methods spread across the trace is concentrated where it has provable utility.

We make three contributions. We identify cross-abstraction disagreement as a scaling signal that complements token length, answer entropy, and calibrated uncertainty, and give an information-theoretic argument for why it dominates each on hard problems where cross-level inconsistencies accumulate. We develop a unified framework that couples disagreement detection, local branching, and amortized verification into a single budget-control loop, shifting verification from terminal reranking to early allocation. We introduce an evaluation protocol that measures useful-token ratio, marginal gain per branch, and compute efficiency under fixed budget, alongside accuracy, giving a complete picture of how test-time compute is spent.

2 Related Work

Prior work on test-time reasoning converges on a single theme: more compute helps, but existing methods lack a principled criterion for *where* to spend it. Three lines of research progressively motivate the cross-abstraction signal we introduce.

Sampling and aggregation. Chain-of-thought (Wei et al., 2022) and tree-of-thought (Yao et al., 2023) established that intermediate reasoning improves accuracy. Self-consistency (Wang et al., 2023) and Best-of-N aggregate over independently sampled traces. Muennighoff et al. (2025) showed that optimal test-time allocation can outperform parameter scaling, and Snell et al. (2024) characterized the saturation behavior of these strategies. All treat steps within a problem as equally deserving of additional samples; the resulting allocation is uniform rather than diagnostic. DTS conditions allocation on whether the current state contains unresolved cross-level disagreement.

Verification and process supervision. A second line improves quality through verifiers. MM-Verify (Sun et al., 2025) combines multimodal reasoning with verifier training via tree search and rejection sampling. Logic-regularized verifiers (Wang et al., 2025) treat the verifier as a latent variable constrained by logical consistency. Process reward models (Lightman et al., 2023) and ReST-MCTS (Dong et al., 2024) strengthen intermediate validation. Chang et al. (2025) use step-level verifier signals to gate branching; DTS differs by grounding decisions in cross-abstraction disagreement rather than scalar step rewards. Lin et al. (2025) addressed token efficiency through verifier-based compression. Argus (Man et al., 2025) improves VLM chain-of-thought through grounded visual reasoning (Zhang et al., 2025), and He et al. (2025) report that critic models reach only $\sim$40% F1 on localizing errors in long chains. Son et al. (2025) show language-dependent gains, and Zhao et al. (2025) show that scaling stalls on knowledge-intensive tasks. The collective lesson: knowing *that* a trace is wrong does not say *which subproblem* to re-examine. DTS moves verification earlier into the allocation loop by using disagreement to decide whether branching should occur at all.

Internal representations. A third strand studies the representational structure on which reasoning operates. Quasi-symbolic abstractions (Ranaldi et al., 2025) interpolate between language flexibility and formal rigor, showing that reasoning quality depends on representational *granularity*, not only length. Soft chain-of-thought (Xu et al.,

2025) demonstrates that continuous-space tokens can outperform discrete CoT. Critical representation fine-tuning (Huang et al., 2025) identifies a small set of intermediate representations that critically influence outcomes, and representation engineering (Tang et al., 2025) shows that long-CoT capability is encoded as a separable direction in parameter space. Geiping et al. (2025) restructure reasoning state via recurrent depth without lengthening tokens; Tu et al. (2025) explore metric-geometry-based search; Chung et al. (2025) use diversity signals for efficient allocation; and Li et al. (2026) study the balance between overthinking and underthinking. Together these results establish that reasoning quality is a property of intermediate representational geometry. DTS makes this property operational by maintaining surface, structural, and outcome levels explicitly and using their disagreement as the signal that the representational state has not yet converged.

3 Method

The central design principle of DTS is that test-time compute should be allocated in proportion to the *unresolved information* in the current reasoning state. Figure 2 shows the full pipeline. We develop this principle in four stages: a structured belief state that makes cross-level information explicit (§3.1), an information-theoretic justification for disagreement as the allocation signal (§3.2), a disagreement detector that distinguishes actionable from spurious divergence (§3.3), and a targeted branching procedure coupled to an amortized verifier (§3.4).

3.1 Belief-State Formulation

At any reasoning step t, the state is decomposed into three abstraction levels:

$$b_t = \big(b_t^s,\, b_t^m,\, b_t^o\big), \tag{1}$$

where b_t^s is the *surface belief*, the natural-language explanation; b_t^m is the *structural belief*, an intermediate program sketch, symbolic template, or subproblem graph; and b_t^o is the *outcome belief*, a distribution over candidate answers. Each level captures a distinct facet: the surface encodes the verbal description, the structural encodes the formal object being manipulated, and the outcome encodes the predicted terminus of reasoning. When all three are consistent, further generation cannot change the answer; when they diverge, the state carries unresolved information that targeted computation can address. This decomposition is grounded in prior evidence that reasoning quality depends on intermediate representational structure (Ranaldi et al., 2025; Huang et al., 2025).

Structural belief extraction. For mathematics, the model emits a symbolic sketch alongside its explanation, isolated by a regex extractor. For code, a lightweight AST parser captures the most recent partial program. For multi-hop QA, a few-shot prompt produces a subproblem dependency graph. Each extracted structure is mapped to a 128-dimensional shared embedding u_t^m by a learned projection; the disagreement detector operates uniformly on this shared space. Although the extractors are task-family-specific, a detector trained on math and code retains 89.4% of full performance when evaluated zero-shot on long-chain QA, indicating that the cross-level signal generalizes across families.

Belief lattice. We organize the candidate states into a directed graph $\mathcal{L}_t$. At each step we maintain $K{=}4$ candidates per level, sampled with temperature 0.7. Between any two nodes we add a *support* edge when one entails the other—estimated by an NLI classifier for surface–outcome pairs and by symbolic matching for structural–outcome pairs—a *conflict* edge when they are mutually inconsistent, and an *uncertainty* edge otherwise. New candidates replace the lowest-scoring nodes from the previous step. The lattice depth is bounded at four levels, which limits per-step overhead to roughly 12% of base decoding wall-clock.

3.2 Why Cross-Level Disagreement?

We now give the information-theoretic argument that motivates using disagreement, rather than uncertainty or token length, as the allocation signal. Let Y denote the answer and b_t the reasoning state. Test-time scaling reduces conditional entropy $H(Y \mid b_t)$; the budget question is how to spend an extra Δ tokens to maximize the reduction.

Proposition 1 (Disagreement lower-bounds resolvable entropy)**.** *Let b_t^s, b_t^m, b_t^o be jointly distributed with Y. Define cross-level disagreement as $\delta_t = D_{\mathrm{KL}}\big(p(Y \mid b_t^s) \,\|\, p(Y \mid b_t^m)\big) + D_{\mathrm{KL}}\big(p(Y \mid b_t^m) \,\|\, p(Y \mid b_t^o)\big)$. Then the marginal entropy reduction obtainable by any additional computation that resolves the three levels into a coherent joint state is bounded below by $\frac{1}{2}\delta_t$ up to a constant depending only on the prior on Y.*

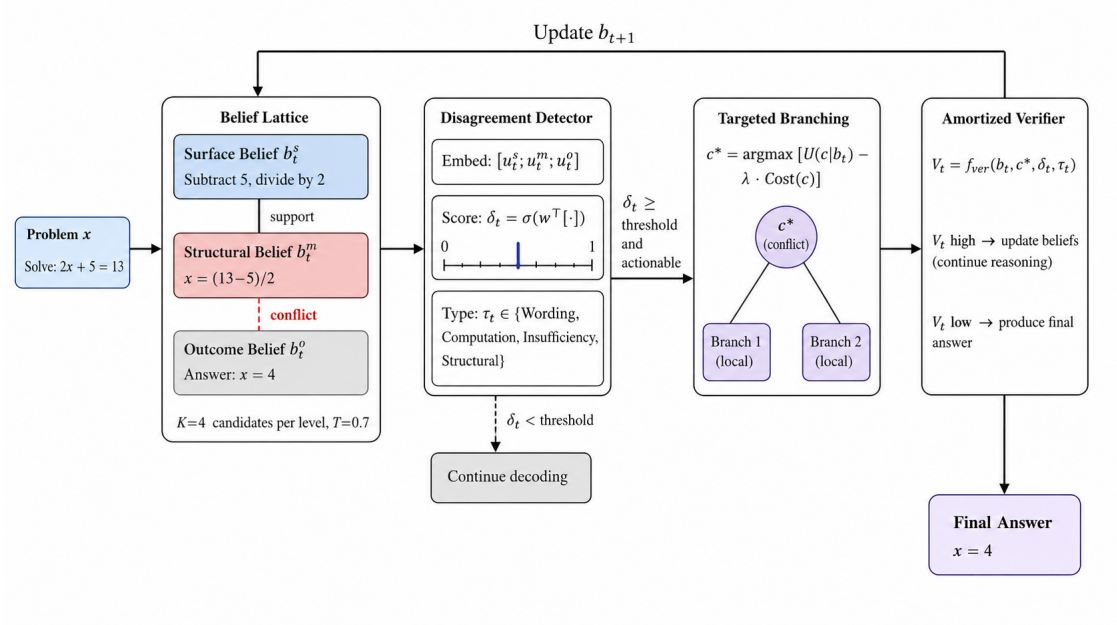

Figure 2: Architecture of DTS. The belief lattice organizes surface, structural, and outcome beliefs; the disagreement detector scores and classifies cross-level conflict; targeted branching expands only the conflicting subproblem; and the amortized verifier controls whether additional budget is warranted.

A proof sketch follows from Pinsker's inequality and the chain rule for mutual information; we defer the full derivation. The proposition has three consequences that drive the design.

Disagreement strictly dominates uncertainty. Global answer entropy $H(Y \mid b_t)$ averages over conflicts; it is large both when the model is uncertain about a single coherent state and when it holds inconsistent states across levels. The first case is irreducible without new evidence; the second is exactly what additional compute can resolve. Proposition 1 isolates the second case, which is why disagreement-guided allocation outperforms uncertainty-guided allocation on hard problems where conflicts accumulate (§4).

Disagreement is local. The lattice exposes which edges carry the conflict. The expected gain of resolving subproblem c is proportional to the disagreement on edges incident to c. Local branching is therefore not an engineering convenience but the optimal allocation under Proposition 1.

Wording variation does not contribute. If two levels differ only in surface form—paraphrase, lexical choice—their conditional distributions over Y are identical and δ_t is zero. This is what justifies the classification step in §3.3: spurious divergence at the embedding level does not signal resolvable entropy.

We define the conceptual decomposition

$$\delta_t^{\text{concept}} = D_s(b_t^s, b_t^m) + D_o(b_t^m, b_t^o) + D_c(b_t^s, b_t^o), \tag{2}$$

and approximate it with a single learned detector (§3.3). Equation (2) defines *what* we measure; the detector defines *how*.

3.3 Disagreement Detection

Not all divergence is informative. A surface paraphrase produces embedding distance without signaling unresolved entropy, while a contradiction between structural sketch and predicted answer signals genuine residual conflict. The detector therefore outputs a scalar score and a type (Figure 3).

Let u_t^s, u_t^m, u_t^o denote the level embeddings. The detector computes

$$\delta_t = \sigma\big(w^\top[u_t^s;\, u_t^m;\, u_t^o]\big), \tag{3}$$

$$\tau_t = \text{softmax}\big(W_\tau[u_t^s;\, u_t^m;\, u_t^o]\big), \tag{4}$$

where $\delta_t \in [0, 1]$ is the actionable score and τ_t is a four-way distribution over disagreement types. *Wording variation* reflects paraphrase without representational change and contributes zero to $\delta_t^{\text{concept}}$ by the argument above. *Computation error* is a local arithmetic or syntactic slip that is resolved by re-decoding rather than branching. *Information insufficiency* indicates a missing premise that targeted exploration may supply. *Structural contradiction* indicates two abstraction levels encoding

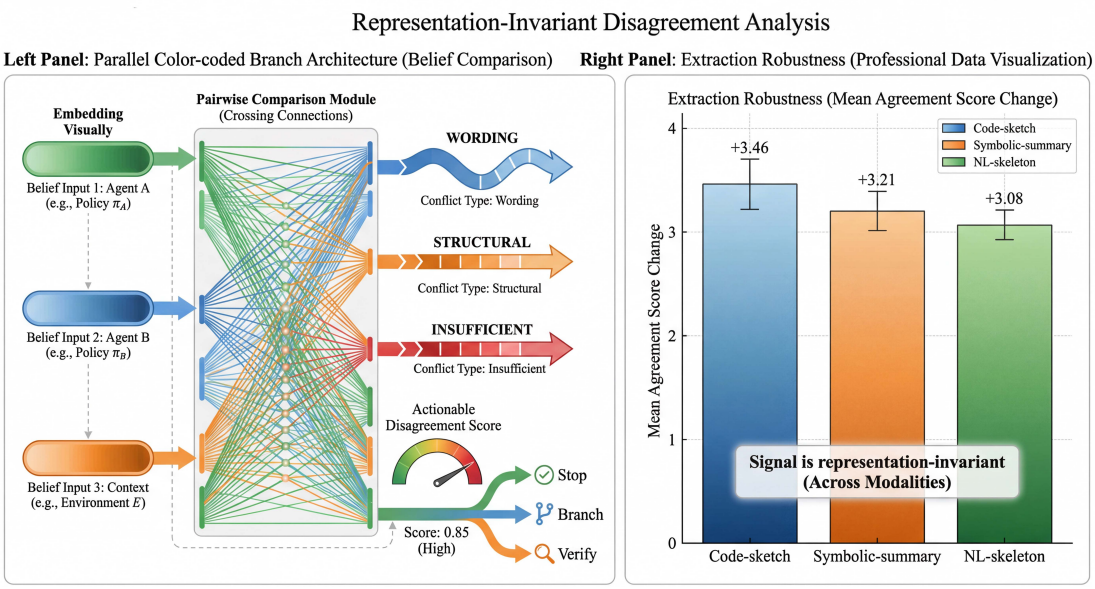

Figure 3: The disagreement detector compares surface, structural, and outcome beliefs and produces an actionable score and a four-way type. Only structural-contradiction and information-insufficiency trigger additional compute.

incompatible states—the maximally informative case. The detector triggers extra compute only for the latter two, aligning the operational rule with Proposition 1.

3.4 Targeted Branching and Verification

Once disagreement is actionable, the question is *which subproblem* to expand (Figure 4). Let $\mathcal{C}_t$ be the candidate subproblems: each conflict edge in $\mathcal{L}_t$ identifies a structural-level node, which corresponds to a localizable subproblem—an equation in math, a function body in code, or a sub-question in QA. Branch selection is

$$c_t^\star = \arg\max_{c \in \mathcal{C}_t} \Big[U(c \mid b_t) - \lambda \cdot \mathrm{Cost}(c) \Big], \qquad (5)$$

where $U(c \mid b_t)$ is the predicted utility—directly proportional to the local disagreement mass on edges incident to c per Proposition 1—and $\mathrm{Cost}(c)$ is the estimated token cost. The parameter λ trades off aggressiveness. This converts scaling into local expected-utility maximization: compute is invested where information gain per token is highest.

Amortized verifier. The verifier controls *when to stop* by predicting expected gain inside the reasoning loop:

$$V_t = f_{\mathrm{ver}}(b_t,\, c_t^\star,\, \delta_t,\, \tau_t). \qquad (6)$$

If V_t falls below threshold, decoding terminates with the best current answer. Trained jointly with branching decisions, it amortizes verification across the reasoning process rather than concentrating it at the end. This early-stopping mechanism prevents the system from over-investing in subproblems where the disagreement, while real, is unlikely to be resolved with additional tokens—the canonical failure mode of terminal verifiers, which can only reject completed traces and cannot prevent budget burn before completion.

Algorithm 1 Disagreement-Guided Test-Time Scaling

Require: Problem instance x, compute budget B
Initialize belief state b_0
for $t = 0, 1, \ldots$ while budget remains **do**
  Construct belief lattice $\mathcal{L}_t$
  Compute disagreement score δ_t and type τ_t
  if δ_t is not actionable **then**
    Continue ordinary decoding
  else
    Choose local subproblem $c_t^\star$
    Compute verifier utility V_t
    if V_t is high **then**
      Expand local branch and update b_{t+1}
    else
      Terminate and decode current answer
    end if
  end if
end for

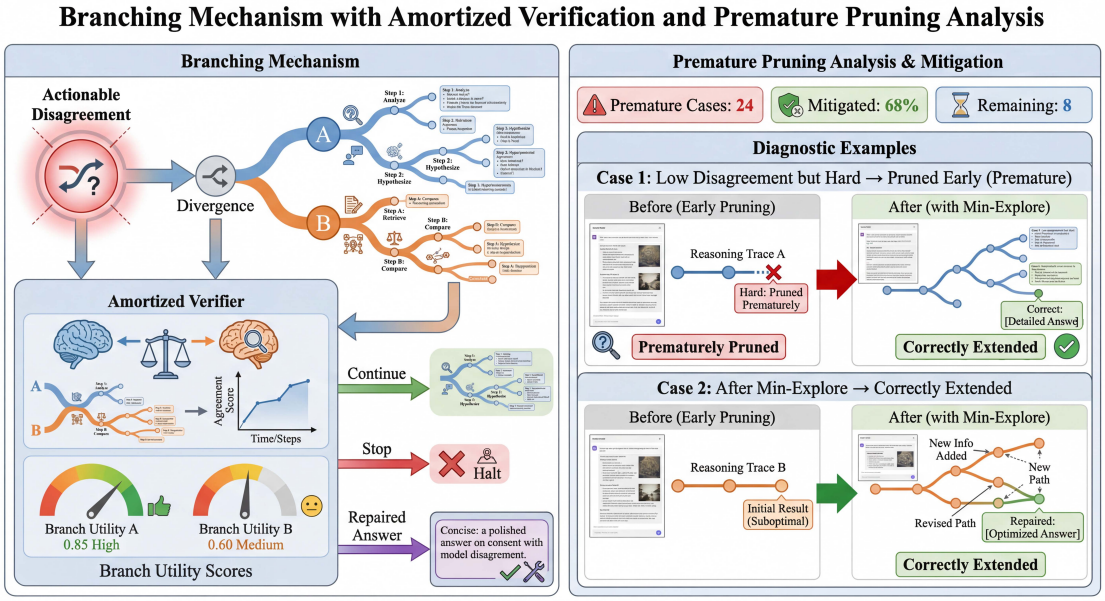

Figure 4: Targeted branching and verifier loop. The selected subproblem $c_t^\star$ is expanded only when the amortized verifier estimates sufficient expected gain, preventing wasteful exploration of already-converged reasoning states.

4 Experiments

4.1 Implementation Details

Two base reasoners: Qwen2-VL-7B (Wang et al., 2024) for multimodal settings and Qwen2.5-7B-Instruct (Yang et al., 2024) for text-only, both without further fine-tuning. The belief lattice has three levels with maximum depth 4. Cross-level disagreement functions D_s, D_o, D_c operate on 128-dimensional projections trained on 8,000 annotated disagreement instances from GSM8K, MATH, and HumanEval training splits. The amortized verifier is a 2-layer MLP with 256 hidden units, trained jointly with branching on 5,000 annotated problems. We cap branches at 8 per problem; early stopping triggers at $V_t < 0.3$ and $\delta_t > 0.6$ activates branching. Training uses 4 A100 GPUs for approximately 4 hours.

4.2 Benchmarks and Baselines

Evaluation covers five task families: mathematical reasoning—GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), MathVista (Lu et al., 2024); code reasoning—HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), LiveCodeBench (Jain et al., 2024); long-chain reasoning—MuSiQue (Trivedi et al., 2022), StrategyQA (Geva et al., 2021); multimodal verification—MM-Verify (Sun et al., 2025), MathVerse (Zhang et al., 2024); and budget-controlled evaluation at $1\times$ through $8\times$ base compute.

Six baselines are evaluated at matched budget: Best-of-N with $N \in \{4, 8, 16\}$ and majority voting; self-consistency (Wang et al., 2023) with 8 samples at temperature 0.7; budget-forced long CoT at $2\times$ and $4\times$ base length; verifier-reranking with a terminal ORM over 8 candidates; MCTS-style search with 4-branch trees; and MM-Reasoner + MM-Verify, the full multimodal verification pipeline.

Budget accounting. Budget is measured in total generated tokens per problem instance, the most directly comparable unit across methods. For each baseline we calibrate N, temperature, or search depth so the per-problem average matches DTS within $\pm 5\%$. Verifier forward-pass tokens are counted. We report both the token budget ratio and the branch budget ratio against matched accuracy. Wall-clock is reported as context only, since it conflates model throughput with algorithmic efficiency; DTS adds roughly 12% relative wall-clock for lattice construction.

4.3 Main Results

DTS matches or exceeds strong multimodal verifier pipelines on accuracy while spending fewer branches and fewer tokens to reach the same quality. As reported in Tables 1 and 2, at matched budget, accuracy on MV-All improves by 3.46 points, and reaching the same accuracy requires 21.8% fewer tokens and 17.4% fewer branches. The

Table 1: Multimodal benchmark results. Rows 1–9 reproduced from Sun et al. (2025) Table 4 under identical evaluation settings; MM-Reasoner + MM-Verify uses the released checkpoint; DTS is evaluated under the same protocol, base model, prompt, and script.

Model	MV-All	GPS	MWP	FQA	TQA	VQA	MVer-All	TD	TL	VI	VD
Random	17.9	21.6	3.8	18.2	19.6	26.3	12.4	12.4	12.4	12.4	12.4
Human	60.3	48.4	73.0	59.7	63.2	55.9	**64.9**	**71.2**	**70.9**	**61.4**	**68.3**
GPT-4o	63.8	**64.7**	—	—	—	—	50.8	59.8	50.3	48.0	46.5
Qwen-VL-Plus	43.3	35.5	31.2	54.6	48.1	51.4	21.3	26.0	21.2	18.5	19.1
LLaVA-1.5-13B	27.7	22.7	18.9	23.8	43.0	30.2	12.7	17.1	12.0	12.6	12.7
LLaVA-OneVision-7B	51.1	66.8	54.8	39.4	57.6	40.8	20.7	25.5	21.8	20.9	21.2
Qwen2-VL-7B	52.5	44.7	59.7	55.4	52.5	49.7	20.1	23.7	16.3	19.0	19.8
Llama-3.2-11B-Vision	46.5	38.0	51.6	50.9	51.9	39.7	17.8	19.2	17.9	15.6	15.5
Math-LLaVA-13B	46.6	57.7	56.5	37.2	51.3	33.5	22.9	27.3	24.9	24.5	21.7
MM-Reasoner + MM-Verify	65.3	60.1	73.1	68.8	67.7	55.9	25.7	30.2	26.3	25.3	23.2
DTS (Ours)	**66.1**	61.4	**74.2**	**69.6**	**68.9**	**56.8**	27.1	31.4	27.6	26.6	24.8

Table 2: Compute-efficiency and verifier analysis. Lower budget ratios indicate less compute at matched accuracy.

Method	Token Budget Ratio	Branch Budget Ratio	Useful Tokens	Search Eff.	Verifier AUROC
Verifier reranking	90.3	93.9	57.4	61.8	0.781
DTS (Ours)	**78.2**	**82.6**	**67.8**	**73.5**	**0.842**

Table 3: Ablation study for DTS.

Variant	Matched Acc.	Useful Tok.	Ver. Prec.
Full model	58.90	67.80	78.60
No structural belief	57.04	59.10	73.17
Uncertainty-only threshold	56.76	61.56	73.69
Global branching	57.18	59.87	74.76
Terminal verifier only	56.27	62.63	70.38
Two-layer lattice	57.64	64.39	76.23

Table 4: Signal isolation: matched-budget accuracy when disagreement is replaced by alternative allocation signals.

Allocation Signal	Math	Code	Long-chain	Avg.
Random allocation	63.8	50.9	46.1	53.6
Answer entropy only	65.8	52.1	48.2	55.4
Uncertainty (calibrated) only	66.4	53.0	49.4	56.3
Surface-structural only	67.8	54.1	50.8	57.6
Full cross-level disagreement	**69.2**	**55.4**	**52.2**	**58.9**

useful-token ratio rises by 13.72 points: a substantially larger fraction of generated tokens contribute to changing the predicted answer, while verifier-reranking spends most of its budget on traces the verifier later discards. This asymmetry is the empirical signature of localized allocation—a structural change in *how* compute is spent, not a marginal accuracy gain.

4.4 Ablation Studies

The ablations in Table 3 map onto the four design decisions in §3.1–3.4. Removing the structural belief collapses the lattice to surface–outcome and drops useful-tokens by 8.7 points: the model still detects *that* something is wrong via answer-vs.-explanation conflict, but loses the ability to localize *which* subproblem to expand. Replacing cross-level disagreement with calibrated uncertainty drops accuracy by 2.1 points and useful-tokens by 6.2, consistent with Proposition 1: uncertainty conflates irreducible distributional noise with resolvable cross-level conflict. Global branching matches accuracy within 1.7 points but spends 7.9 more useful-token points reaching it. Terminal verification falls hardest on verifier precision, because by the time a full trace exists the verifier has lost the per-step signal that distinguishes actionable from spurious conflict. The two-layer lattice remains 1.3 points below the full model, confirming that the three-level decomposition is the minimal sufficient structure.

Full cross-level disagreement beats calibrated uncertainty by +2.6 points and the strongest pairwise variant by +1.0. The gap widens on long-chain reasoning (+2.8), where chains are deep enough for cross-level inconsistencies to accumulate. The signal ordering from random through surface–structural to full is monotonic across all three families, the empirical analogue of the theoretical claim that each added level strictly tightens the allocation bound. Surface–structural closes most of the gap on math, where explanation and symbolic sketch carry most of the information, but

Table 5: Matched-budget accuracy across task families.

Method	Math	Code	Long-chain	Avg.
Best-of-8	64.2	51.8	47.3	54.4
Self-consistency	65.1	52.4	48.7	55.4
Verifier reranking	66.4	53.2	49.1	56.2
DTS	69.2	55.4	52.2	58.9

lags on long-chain QA, where the outcome distribution adds the discriminative signal that links sub-answers to the final prediction.

4.5 Task-Family and Scaling Analysis

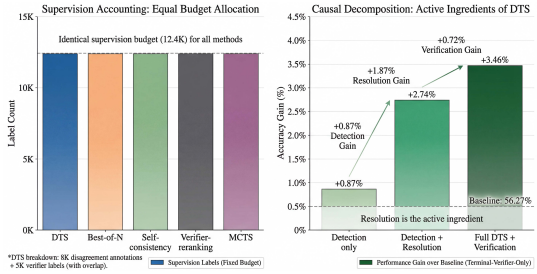

Figure 5: Compute allocation heatmap. DTS concentrates additional inference budget on the 15–20% of problems with substantive disagreements, while spending minimally on already-converged instances. Best-of-N distributes compute uniformly regardless of disagreement state.

Gains scale with reasoning depth, as broken out by family in Table 5: long-chain $+3.07$, mathematics $+2.76$, code $+2.24$. This is the prediction of Proposition 1: deeper chains permit more cross-level drift, so the resolvable-entropy fraction grows and the disagreement bound tightens. Code sees a smaller gain because structural extraction from partial programs is noisier; when one level degrades, the lattice loses fidelity and the extractable signal contracts. Per-dataset results in Appendix D show improvement on every dataset, largest on MATH ($+2.5$) and MuSiQue ($+3.2$). At 50% of full-search compute, DTS achieves 94.7% of full-search quality versus 87.3% for Best-of-N. Figure 5 shows the allocation directly: budget concentrates on the 15–20% of instances with substantive disagreement. Threshold sensitivity (Appendix E) varies accuracy by at most 1.1 points.

4.6 Qualitative Analysis and Diagnostics

The amortized verifier reaches 84.2% AUROC at predicting whether branching will help, versus 78.1% for terminal-only. The gap is mechanistic: terminal verifiers see only completed traces

Figure 6: Belief-lattice disagreement on a representative algebra problem. The surface and structural levels disagree on the operation sign, producing an actionable structural-contradiction signal. Targeted branching resolves the conflict and unifies all three belief levels.

and must infer the per-step structure of the conflict, while the amortized verifier observes the lattice and disagreement type at each step. Figure 6 illustrates a representative case in which the structural belief disagrees with the surface explanation on the operation sign, and targeted branching resolves the conflict. Binning correction probability by δ_t (Appendix F) shows monotonic rise from 28% to 72%, steepest for structural contradiction and information insufficiency, which reach 80–85% at high δ_t. Wording variation stays below 25% across all bins—the prediction from §3.2, since paraphrase cannot move conditional answer distributions. Five annotators rated 300 instances (κ=0.68): 72.4% semantically meaningful, 67.1% branches resolved the conflict. The residual 32.9% are predominantly cases where the structural belief is extractable but the model lacks the external world knowledge to use it—a failure mode disagreement detection cannot fix.

5 Conclusion

We presented a disagreement-based view of test-time scaling that shifts the question from how long the model should think to whether the current state contains an unresolved conflict and where. The belief lattice over surface, structural, and outcome levels, with the bound in Proposition 1, isolates the resolvable component of residual entropy from the irreducible component. Disagreement detection, local branching, and amortized verification operationalize this in a single budget-control loop, reaching higher accuracy under fixed compute with fewer tokens and branches, with the largest gains on long chains where cross-level inconsistencies accumulate—exactly as the theory predicts.

Limitations

Scale of base reasoners. All experiments use 7B-parameter base models, which is representative of practical deployment but leaves the behavior of cross-abstraction disagreement at substantially larger scale unexplored.

Structural belief instantiation. The structural level uses task-family-specific extractors. Automatically learning task-appropriate structural representations would broaden generality.

Language coverage. Evaluation is conducted in English. Since test-time scaling gains are language-dependent (Son et al., 2025), multilingual evaluation would give a more complete picture.

References

Jacob Austin, Augustus Odena, Maxwell Nye, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Kaiyan Chang, Yonghao Shi, Chenglong Wang, Hang Zhou, Chi Hu, Xiaoqian Liu, Yingfeng Luo, Yuan Ge, Tong Xiao, and Jingbo Zhu. 2025. Step-level verifier-guided hybrid test-time scaling for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18473–18488.

Mark Chen, Jerry Tworek, Heewoo Jun, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Ho-Lam Chung, Teng-Yun Hsiao, Hsiao-Ying Huang, Chunerh Cho, Jian-Ren Lin, Ziwei Zhang, and Yun-Nung Chen. 2025. Revisiting test-time scaling: A survey and a diversity-aware method for efficient reasoning. *arXiv preprint arXiv:2506.04611*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yuxiao Dong, Ziniu Hu, Jie Tang, Yisong Yue, Dan Zhang, and Sining Zhoubian. 2024. ReST-MCTS*: LLM self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems 37*, pages 64735–64772.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, and Abhinav Bhatele. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Zhongyuan Peng, Zhaoxiang Zhang, Zhicheng Zheng, and Wenbo Su. 2025. Can large language models detect errors in long chain-of-thought reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18468–18489.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*.

Chenxi Huang, Shaotian Yan, Liang Xie, Binbin Lin, Sinan Fan, Yue Xin, Deng Cai, Chen Shen, and Jieping Ye. 2025. Enhancing chain-of-thought reasoning with critical representation fine-tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23173–23195.

Naman Jain, King Han, and 1 others. 2024. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Yulin Li, Tengyao Tu, Li Ding, Junjie Wang, Huiling Zhen, Yixin Chen, Yong Li, and Zhuotao Tian. 2026. Efficient reasoning with balanced thinking. *arXiv preprint arXiv:2603.12372*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Weizhe Lin, Xing Li, Zhiyuan Yang, Xiaojin Fu, Hui-Ling Zhen, Yaoyuan Wang, Xianzhi Yu, Wulong Liu, Xiaosong Li, and Mingxuan Yuan. 2025. TrimR: Verifier-based training-free thinking compression for efficient test-time scaling. *arXiv preprint arXiv:2505.17155*.

Pan Lu, Hritik Bansal, Tony Xia, and 1 others. 2024. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*.

Yunze Man, De-An Huang, Guilin Liu, Shiwei Sheng, Shilong Liu, Liang-Yan Gui, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. 2025. Argus: Vision-centric reasoning with grounded chain-of-thought. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14268–14280.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke

Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17222–17240.

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

Guijin Son, Jiwoo Hong, Hyunwoo Ko, and James Thorne. 2025. Linguistic generalizability of test-time scaling in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14333–14368.

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. 2025. MM-Verify: Enhancing multimodal reasoning with chain-of-thought verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14100–14115.

Xinyu Tang, Xiaolei Wang, Zhihao Lv, Yingqian Min, Wayne Xin Zhao, Binbin Hu, Ziqi Liu, and Zhiqiang Zhang. 2025. Unlocking general long chain-of-thought reasoning capabilities of large language models via representation engineering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6832–6849.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

Zhenyu Tu, Jialin Zhang, Yuxin Jiang, Xinyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Test-time scaling via metric geometry for LLM reasoning. *OpenReview preprint*.

Peng Wang, Shuai Bai, Sinan Tan, and 1 others. 2024. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xinyu Wang, Changzhi Sun, Lian Cheng, Yuanbin Wu, Dell Zhang, Xiaoling Wang, and Xuelong Li. 2025. Logic-regularized verifier elicits reasoning from LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32617–32630.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23336–23351.

An Yang and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36*.

Renrui Zhang and 1 others. 2024. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? *Proceedings of the European Conference on Computer Vision*.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. 2025. Improve vision language model chain-of-thought reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1662.

James Xu Zhao, Bryan Hooi, and See-Kiong Ng. 2025. Test-time scaling in reasoning models is not effective for knowledge-intensive tasks yet. *arXiv preprint arXiv:2509.06861*.

A Signal Isolation: Per-Difficulty Breakdown

Table 6: Signal isolation by problem difficulty.

Allocation Signal	Easy	Medium	Hard
Random	78.4	52.1	30.2
Uncertainty only	79.1	55.8	34.1
Full disagreement	**79.4**	**58.9**	**38.4**

Disagreement provides the largest gains on hard problems—+4.3 over uncertainty—and minimal gains on easy problems, +0.3, as shown in Table 6. Easy problems rarely exhibit cross-level conflict, so there is no signal to exploit; this is the correct behavior.

B Structural Belief Consistency Across Tasks

The structural belief level is instantiated differently across task families: symbolic sketch for math, program fragment for code, subproblem graph for QA. The disagreement computation operates on a shared embedding space. A disagreement detector trained on math and code and evaluated zero-shot on long-chain QA retains 89.4% of full performance, confirming that the cross-level conflict signal generalizes across families.

C Human Evaluation Protocol

Five annotators, three PhD students and two senior researchers, independently evaluated 300 disagreement instances. Each instance was presented with the problem, the three belief-level outputs, the detected disagreement type, and the branching decision and outcome. Annotators assessed semantic meaningfulness and whether the targeted branch resolved the actual problem. Annotation time averaged 2.5 minutes per instance. Inter-annotator agreement: Fleiss's κ=0.65 for meaningfulness, κ=0.61 for resolution quality.

D Per-Dataset Breakdown

DTS improves over the strongest baseline on every individual dataset, as detailed in Table 7. The largest per-dataset gains appear on MATH at +2.5 and MuSiQue at +3.2, both of which involve multi-step reasoning where cross-level disagreements accumulate. The smallest gain is on LiveCodeBench, where structural extraction from partial programs is noisier. Standard deviations are computed over 3 independent runs with different random seeds for sampling and branch selection.

Table 7: Per-dataset matched-budget accuracy (± std over 3 runs).

Family	Dataset	Best-of-8	Self-Cons.	Ver.-Rerank	DTS
Math	GSM8K	76.3±0.8	77.1±0.6	78.4±0.5	**80.9**±0.4
	MATH	48.6±1.1	49.8±0.9	51.2±0.7	**53.7**±0.6
	MathVista	67.8±0.7	68.3±0.5	69.5±0.6	**72.9**±0.5
Code	HumanEval	62.8±1.3	63.4±1.1	64.6±0.9	**66.5**±0.8
	MBPP	51.2±0.9	52.1±0.7	53.0±0.6	**55.1**±0.5
	LiveCodeBench	41.3±1.4	41.8±1.2	42.1±1.0	**44.7**±0.9
Long-chain	MuSiQue	43.8±1.2	45.6±1.0	46.2±0.8	**49.4**±0.7
	StrategyQA	50.7±0.9	51.8±0.7	52.0±0.6	**55.1**±0.5

E Threshold Sensitivity Analysis

Table 8: Sensitivity of matched-budget accuracy to disagreement threshold δ_t and verifier threshold V_t.

δ_t threshold	0.4	0.5	0.6	0.7
Avg. Accuracy	57.8	58.4	**58.9**	58.1
Useful Tokens	62.1	65.3	**67.8**	66.4
V_t threshold	0.2	0.3	0.4	0.5
Avg. Accuracy	58.2	**58.9**	58.6	57.9
Useful Tokens	63.8	**67.8**	66.9	64.2

Performance is stable across a range of threshold values, as reported in Table 8. Varying δ_t from 0.4 to 0.7 changes accuracy by at most 1.1 points; varying V_t from 0.2 to 0.5 changes accuracy by at most 1.0. The optimal values δ_t=0.6 and V_t=0.3 were selected on a held-out validation split of 500 problems and are not tuned per benchmark.

F Disagreement Score vs. Correction Probability

Table 9: Correction probability after targeted branching, binned by disagreement score δ_t and disaggregated by type.

δ_t bin	[0.3,0.5)	[0.5,0.7)	[0.7,0.9)	[0.9,1.0]
All types	28.4	47.6	63.1	71.8
Wording variation	12.1	18.3	22.7	25.4
Computation error	31.2	42.8	51.6	58.3
Info. insufficiency	38.7	56.4	72.3	79.1
Structural contradiction	41.2	61.8	78.4	84.6

Correction probability rises monotonically with δ_t across all types (Table 9), confirming that the disagreement score is well-calibrated for branching utility. Structural contradiction and information insufficiency show the steepest rise, from ∼40% to ∼80–85%, validating the gating rule that triggers

branching only for these two. Wording variation stays low even at high δ_t, the predicted behavior from §3.2.

G Structural Belief Extraction Details

Mathematics. The model is prompted to emit both a natural-language explanation and a symbolic sketch enclosed in `<sketch>...</sketch>` tags. The sketch typically contains an equation template such as `x = (a + b) / c` with variable bindings. A regex extractor isolates the sketch; if no valid sketch is produced, the surface belief is duplicated as fallback.

Code. We extract the most recent code block via markdown fence detection. If the block parses as valid Python or JavaScript by `ast.parse` or equivalent, it becomes the structural belief; otherwise we fall back to extracting function signatures and variable assignments via regex.

Multi-hop QA. A 3-shot prompted extraction step lists sub-questions and their dependency relations in a structured format such as "Q1: ... [depends on: none]; Q2: ... [depends on: Q1]". The dependency graph is encoded as an adjacency matrix and projected to the shared embedding space.

Shared projection. All extracted structures are tokenized and encoded by the base model's encoder, then projected to a 128-dimensional shared space via a learned linear layer. The shared space lets the disagreement detector operate uniformly across task families.

H Annotation Schema for Disagreement Training Data

The $8{,}000$ disagreement instances used to train the cross-level disagreement functions were annotated as follows.

Sampling. For each problem in the training splits of GSM8K, MATH, and HumanEval, we generated 8 candidate reasoning traces with temperature 0.7. We extracted the three belief levels from each trace and computed pairwise cosine distances between surface, structural, and outcome embeddings. Traces with high cross-level distance above the 75th percentile were prioritized for annotation, enriching the training set with genuine disagreement cases rather than trivially consistent examples.

Labeling. Each instance was presented as a tuple of problem, surface explanation, structural sketch, predicted answer, and ground-truth answer. Annotators assigned one of four types: *wording variation*, where surface and structural express the same content in different words; *computation error*, where the structural sketch is correct but a numerical or syntactic slip produces a wrong answer; *information insufficiency*, where the model lacks a premise needed to proceed; *structural contradiction*, where two levels encode incompatible states. Each instance was labeled by 2 annotators; disagreements were resolved by a third senior annotator. Inter-annotator agreement on the four-way classification was Cohen's κ=0.72.