# Escaping Reasoning Basin Collapse with History-Biased Search

**Lu Cheng**
Department of Computer Science and Engineering
Penn State University
University Park, PA 10681
lqc5822@psu.edu

## Abstract

Inference-time search with large language models (LLMs) often concentrates on a small set of structurally or semantically similar trajectories, leaving alternative reasoning strategies underexplored—a failure mode we call *reasoning basin collapse*. We introduce BASIN, a training-free, history-biased search method that groups reasoning states into basins and accumulates a revisit penalty on repeatedly selected basins, reallocating a fixed inference budget toward underexplored reasoning strategies. Under matched inference budgets, BASIN improves over Tree of Thoughts (ToT) by up to $+22$pp on Game of 24 and $+6.7$pp on MuSR. Because indiscriminate diversification can over-explore once search has found a promising basin, we further introduce QA-BASIN, a quality-aware variant that weakens the revisit penalty for high-quality basins and yields more robust gains. To characterize when basin-aware search helps, we introduce the *redundancy gap* $\Delta$, which measures the difference in search concentration between correct and incorrect predictions: standard ToT often operates near $\Delta \approx 0$, whereas BASIN consistently shifts $\Delta$ positive. Together, these results identify reasoning basin collapse as a failure mode of inference-time search and show that history-dependent bias provides a simple, training-free mechanism for escaping redundant reasoning under fixed compute. Code is available at `https://github.com/GitHubLuCheng/basin`.

## 1 Introduction

When faced with a difficult reasoning problem, a careful human solver rarely repeats the same line of argument indefinitely—instead, they try genuinely different explanations before producing more variants of the same one. Yet this discipline is not built into most LLM inference-time search procedures. Recent work improves LLM reasoning by scaling inference-time compute, generating multiple candidate trajectories and selecting or aggregating among them [Wei et al., 2022, Wang et al., 2023, Yao et al., 2023, Besta et al., 2024, Hao et al., 2023, Zhou et al., 2024, Ding et al., 2025]. More trajectories, however, do not necessarily imply more distinct reasoning strategies. Standard search is largely unaware of which candidates revisit an already explored strategy and can therefore spend substantial inference budget producing variations of the same underlying approach. We call this failure mode **reasoning basin collapse**.

Figure 1 illustrates this phenomenon on MuSR [Sprague et al., 2024], a benchmark requiring multi-step reasoning. Under standard Tree of Thoughts (ToT), the effective basin count $N_{\text{eff}}$ (formally defined in Sec. 4.1) falls well below the number of generated states: Although ToT generates many reasoning states, its effective basin count is only 38% of the number of generated nodes on average, indicating that much of the search budget is spent revisiting a small set of strategies. Moreover, this collapse is *indiscriminate*: incorrect searches can repeatedly revisit wrong strategies just as readily as

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

correct searches can concentrate on useful ones. Search concentration alone therefore provides little indication of whether the explored basin is actually productive.

To formalize this behavior, we define a **reasoning basin** as an equivalence class of reasoning states that pursue the same underlying strategy. For tasks with explicit structure, basins can be defined deterministically from task-relevant symbolic or structural features. For open-ended reasoning, basins can instead be defined semantically, using an extracted central hypothesis and natural language inference (NLI) [Balamurali and Cheng, 2025] to group equivalent hypotheses. This abstraction separates diversity in underlying reasoning strategies from surface-level variation among trajectories and gives search an explicit notion of where it has already explored.

**BASIN.** We propose BASIN: **B**asin-**A**ware **S**earch for **I**nference-time Reaso**N**ing, a training-free *history-biased search* principle for inference-time LLM reasoning. The design is inspired by metadynamics [Laio and Parrinello, 2002, Tiwary and Parrinello, 2013] in molecular dynamics [Wang et al., 2020], an enhanced-sampling method that discourages a physical system from repeatedly visiting previously explored configurations by accumulating a history-dependent bias. BASIN transfers this principle to reasoning search by tracking how often each reasoning basin has been selected and applying a logarithmic revisit penalty to candidates from over-visited basins, progressively reallocating a fixed inference budget toward underexplored strategies while still allowing strong candidates from familiar basins to remain competitive. Because this history bias can over-explore once search has already found a promising basin, we further introduce QA-BASIN, which weakens the revisit penalty for high-quality basins to better balance exploration and exploitation. Both methods act only at candidate selection, require no training or changes to the underlying generator, and can be incorporated into search procedures that repeatedly select among candidate states.

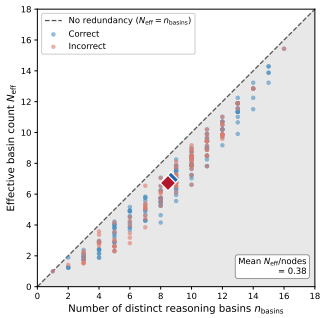


Figure 1: **Reasoning basin collapse in ToT.** Each point is one MuSR problem ($n$=300, `gpt-oss-120b`). The $y$-axis shows $N_{\text{eff}}$, the effective number of basins explored after accounting for visit concentration. All points fall below the diagonal (shaded), where $N_{\text{eff}} < n_{\text{basins}}$, indicating that visits concentrate on previously explored basins.

**Empirical findings.** Across various reasoning tasks, history-biased basin-aware selection improves search under matched inference budgets. However, greater basin diversity does not always imply higher accuracy: BASIN helps most when search is trapped in redundant incorrect regions, but can over-explore after finding a strong basin. QA-BASIN mitigates this exploration–exploitation trade-off by preserving high-quality basins, yielding robust gains across models, tasks, and search frameworks. We also introduce the *redundancy gap* $\Delta$, which measures the difference in search concentration between correct and incorrect predictions. Standard ToT often operates near $\Delta \approx 0$, indicating that correct and incorrect searches exhibit similar levels of concentration, whereas BASIN consistently shifts $\Delta$ positive.

Our contributions are threefold. First, we formalize **reasoning basins** and identify **reasoning basin collapse** as a failure mode of inference-time search. Second, we introduce **BASIN**, a training-free history-biased search method, together with its quality-aware variant **QA-BASIN**, and show improvements across multiple domains, models, and search frameworks under fixed inference budgets. Third, we introduce the **redundancy gap** $\Delta$ as a diagnostic for characterizing harmful search redundancy and motivating more adaptive history-biased reasoning search.

## 2 Related Work

**Reasoning with intermediate steps.** Chain-of-thought (CoT) prompting elicits step-by-step reasoning before the final answer [Wei et al., 2022]; Zero-shot-CoT shows that a simple "think step by step" prompt can elicit reasoning without demonstrations [Kojima et al., 2022]. Further methods improve intermediate rationales through structured decomposition, automatic exemplar selection, and complexity-based sampling [Zhou et al., 2023, Zhang et al., 2023, Zelikman et al., 2022, Fu et al., 2023]. Self-Consistency aggregates multiple sampled traces by majority vote [Wang et al., 2023]. Recent work also explores reasoning in continuous latent space to improve the flexibility and efficiency of intermediate computation [Liu et al., 2026]. These methods improve how reasoning trajectories are generated, represented, or aggregated, but do not explicitly track whether search repeatedly revisits the same underlying reasoning strategy.

**Search-based reasoning.** ToT frames LLM reasoning as search over intermediate states [Yao et al., 2023], with extensions to graph-structured reasoning, planning-style procedures, and agentic tree search [Besta et al., 2024, Hao et al., 2023, Zhou et al., 2024]. Recent work also improves search efficiency: Dynamic Parallel Tree Search reduces redundant exploration in ToT-style inference [Ding et al., 2025], while Policy-Guided Tree Search learns a controller for expansion, branching, and backtracking [Li, 2025]. Related work adaptively controls inference compute through early stopping or routing between models with different reasoning capabilities [Chen et al., 2023, Zhou et al., 2026, Su et al., 2026]. BASIN is complementary to these approaches: rather than deciding how long to search, how to expand the tree, or which model to invoke, it introduces a history-dependent selection bias based on which reasoning strategies have already been explored. By grouping states into reasoning basins and penalizing repeated visits, BASIN directly targets reasoning basin collapse during candidate selection.

**Reflection, refinement, and verification.** Self-Refine iteratively improves outputs through feedback and revision [Madaan et al., 2023, Zhou and Cheng, 2026a,b], while Reflexion uses verbal self-reflection to guide future attempts [Shinn et al., 2023, Huang et al., 2025]. Verifier-based approaches rerank candidates using learned or external evaluators [Lightman et al., 2023], and recent test-time methods study adaptive allocation of reasoning effort [Ling et al., 2026, Zhou et al., 2026, Su et al., 2026]. These methods primarily improve trajectory quality, search control, or inference allocation. BASIN instead controls how search effort is distributed across underlying reasoning strategies using search history; QA-BASIN further conditions this history bias on quality so that promising basins remain competitive while repeatedly visited, lower-quality basins are discouraged.

**Diversity-promoting and history-biased search.** Diverse Beam Search discourages near-duplicate candidates by adding a diversity penalty across beam groups [Vijayakumar et al., 2016]. At the reasoning level, Diversity of Thought elicits distinct prompt-level solution approaches [Naik et al., 2023], while Diversity of Thoughts for agents reduces redundant reflections to broaden decision-space exploration [Lingam et al., 2025]. BASIN shares the goal of reducing redundant search but differs in both the unit of diversity and the mechanism used to enforce it. Rather than maximizing instantaneous diversity among candidates, it defines equivalence classes of states at the *strategy level* and accumulates a history-dependent penalty as the same basin is revisited. This turns diversity from a local property of the current candidate set into a history-aware search signal over previously explored reasoning strategies.

## 3 Method

BASIN is a history-biased modification to inference-time reasoning search. Standard search scores candidate states largely independently of where search has already spent its compute, even when several candidates pursue the same underlying reasoning strategy. As a result, substantial inference budget can be spent repeatedly exploring equivalent trajectories. BASIN makes this search history explicit by grouping states into *reasoning basins* and accumulating a revisit penalty over repeatedly selected basins.

### 3.1 Reasoning Basins

A **reasoning basin** is an equivalence class of states that share the same underlying reasoning strategy. Basin membership captures redundancy relevant to search rather than surface similarity: states belong to the same basin when they pursue the same core hypothesis or induce the same relevant continuation structure, even if their textual realizations differ.

We define a basin assignment function $\mathcal{B} : \mathcal{S} \to \mathcal{Z}$ mapping each reasoning state $s$ to a discrete basin identifier. States satisfying $\mathcal{B}(s) = \mathcal{B}(s')$ are treated as repeated exploration of the same strategy. The basin representation is task-dependent: when explicit structure is available, we use deterministic structural definitions; for open-ended reasoning, we approximate strategy equivalence semantically.

**Structural basins.** For arithmetic tasks such as Game of 24, we define basin membership using the ordered sequence of operations applied so far and the sorted remaining values:

$$\mathcal{B}(s) = \big(\mathrm{ops}(s), \mathrm{sort}(\mathrm{remaining}(s))\big). \tag{1}$$

Here $\mathrm{ops}(s)$ is the ordered operator sequence (e.g., $(\texttt{mul}, \texttt{sub})$), so states applying the same operations in a different order remain distinct. For example, the partial step `11,-,1,=,10` on input $1, 11, 11, 13$

yields basin `sub,|,10,11,13`. This representation captures task-relevant structural redundancy with lightweight deterministic parsing.

**Semantic basins.** For open-ended tasks such as MuSR, exact structural keys are unavailable and string matching is too brittle. We therefore extract from each reasoning trace a `main_hypothesis`, a one-sentence summary of its central claim, and group states that predict the same answer and express compatible hypotheses under an NLI model. Concretely, states $s$ and $s'$ are grouped when

$$\mathrm{ans}(s) = \mathrm{ans}(s') \ \wedge \ \mathrm{NLI}_{\mathrm{ent}}(h_s, h_{s'}) \geq \tau_e \ \wedge \ \mathrm{NLI}_{\mathrm{con}}(h_s, h_{s'}) \leq \tau_c. \tag{2}$$

where $h_s$ denotes the extracted hypothesis and $\tau_e, \tau_c$ control clustering granularity. We use NLI rather than embedding similarity because reasoning traces often share substantial narrative context despite supporting different hypotheses; NLI more directly captures propositional compatibility. We analyze sensitivity to the extractor, NLI model, and thresholds in Appendix G.

### 3.2 History-Biased Basin Selection

Let $\mathrm{visits}[z], z \in \mathcal{Z}$ denote the number of times basin $z = \mathcal{B}(s)$ has previously been selected into the active search set, and let $f(s)$ be the base score assigned by the underlying search procedure (e.g., model likelihood, a value estimate, or a heuristic score). Inspired by metadynamics [Laio and Parrinello, 2002] in molecular dynamics, which uses a history-dependent bias to discourage repeated visits to previously explored regions, BASIN replaces $f(s)$ with

$$\tilde{f}(s) = f(s) - \lambda \log\left(1 + \mathrm{visits}[z]\right). \tag{3}$$

where $\lambda \geq 0$ controls the strength of the history bias. Visit counts are updated after each selection step. Because all states in the same basin share a visit count, the penalty accumulates at the strategy level rather than independently for individual trajectories. Unvisited basins incur no penalty, while repeated visits receive an increasing but sublinear penalty. The resulting search is therefore biased by its own history: as a basin is revisited, candidates from that basin become progressively less competitive, shifting selection toward underexplored strategies. Importantly, revisits are not forbidden; a previously explored basin remains selectable whenever its base-score advantage exceeds the accumulated penalty.

The selection rule is agnostic to how basin membership is constructed: deterministic structural keys are preferable when available, while semantic clustering provides a fallback when no exact task-specific equivalence relation exists.

### 3.3 Quality-Aware BASIN

History-biased exploration introduces an exploration–exploitation trade-off. Penalizing revisits is useful when search is trapped in a repeatedly explored weak basin, but can also redirect compute away from a promising basin simply because it has been visited frequently. When a meaningful quality signal is available, we therefore introduce QA-BASIN:

$$\tilde{f}(s) = f(s) - \lambda \log\left(1 + \mathrm{visits}[z]\right)\left(1 - q_z\right), \tag{4}$$

where $q_z \in [0, 1]$ is the running mean quality score for basin $z$. The quality term modulates the accumulated history bias: high-quality basins receive a weaker revisit penalty, whereas low-quality basins approach the original BASIN penalty. QA-BASIN therefore preserves promising strategies while continuing to discourage redundant exploration of weaker ones. Its effectiveness depends on the verifier providing a meaningful quality signal.

### 3.4 Instantiation in ToT

We instantiate history-biased basin selection within ToT, although the principle applies to any inference-time search procedure that maintains candidate states and repeatedly selects among them. We also evaluate the same mechanism with Graph of Thoughts (Appendix D) and UCT-based MCTS (Sec. 4.3). Standard ToT maintains a beam $\mathcal{A}_t$ of $k$ active states. At each round, it expands the active states into candidate continuations, scores them using $f$, and retains the top $k$. BASIN changes only this selection step: each candidate is assigned a basin through $\mathcal{B}$, rescored using Eq. (3) (or Eq. (4) for QA-BASIN), and ranked by the resulting history-biased score. Visit counts are then updated for the selected states. All other components of the search remain unchanged. Thus, BASIN changes not how candidate reasoning states are generated, but *where search allocates inference compute* as a function of its exploration history.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** Our primary experiments use two complementary benchmarks spanning symbolic and natural-language reasoning. *Game of 24* requires combining four integers using basic arithmetic ($+, -, \times, \div$) to obtain 24; we use the standard 100-problem set from [Yao et al., 2023]. Solutions are verified exactly by evaluating the expression and checking number usage. ***MuSR*** [Sprague et al., 2024] is a multi-step reasoning benchmark covering murder mystery, object placement, and team allocation; we use 300 problems sampled uniformly across subtasks.

To test broader generalization, we additionally evaluate *HumanEval* [Chen et al., 2021], *GSM-Hard* [Gao et al., 2022], and the *Logical Deduction* subtask of BIG-Bench Hard [Srivastava et al., 2022]. HumanEval tests program synthesis and admits deterministic structural basin definitions, while GSM-Hard tests challenging mathematical reasoning. Together, these benchmarks span symbolic search, natural-language reasoning, mathematics, and code generation.

**Models.** Our primary Game of 24 experiments use `gpt-4o-mini` [OpenAI, 2024] and `Qwen3-27B` [Yang et al., 2025]; MuSR uses `gpt-4o-mini` and `gpt-oss-120b` [OpenAI et al., 2025]. For broader evaluation, we additionally test `Qwen2.5-7B-Instruct` [Team, 2024] and `Llama-3.3-70B-Instruct` [Dubey et al., 2024], covering multiple model families and scales.

**Search and Basins.** We use ToT as the primary controlled search framework. BASIN changes only candidate selection through Eq. (3); QA-BASIN uses Eq. (4). Generation, search budget, and final-answer selection are otherwise held fixed. Game of 24 and HumanEval use deterministic structural basin definitions. For MuSR, we extract a `main_hypothesis` from each reasoning trace and construct semantic basins as described in §3.1. The same extraction procedure is applied when computing basin statistics for baseline and basin-aware searches. Under our nine-round MuSR setup, both conditions use 18 generation and 18 hypothesis extraction calls per problem. The Appendices F-G study the sensitivity to the extractor and NLI clustering choices.

**Hyperparameters.** For Game of 24, we use beam size $k$=5, branching factor $b$=5, and depth $T$=3. For MuSR, we use beam size $k$=2 and nine reasoning rounds. Semantic clustering uses entailment threshold $\tau_e$=0.45 and contradiction ceiling $\tau_c$=0.3, chosen to require moderate positive support between hypotheses while excluding pairs with substantial contradictory evidence. Appendix G shows that the results are robust to alternative entailment thresholds and semantic basin constructions. Unless otherwise stated, $\lambda$=3.0. Sampling temperatures are 0.7 for Game of 24 and 0.8 for MuSR and are held fixed across methods. Appendix K provides compute and implementation details. Appendix J shows prompt templates.

**Evaluation metrics.** We report accuracy as the primary performance metric and Pass@k when the final beam can contain multiple candidate answers. To characterize search structure, we report the number of visited basins $n_{\text{basins}}$ and the effective basin count $N_{\text{eff}} = \exp(-\sum_z p_z \log p_z)$, where $p_z$ is the fraction of selected states assigned to basin $z$. $N_{\text{eff}}$ equals the number of basins under uniform visitation and decreases as search concentrates on a subset of them, thereby capturing both basin coverage and the evenness of search allocation. We define redundancy as $\rho = n_{\text{basins}} - N_{\text{eff}}$ and the redundancy gap as

$$\Delta = \mathbb{E}[\rho \mid \text{correct}] - \mathbb{E}[\rho \mid \text{incorrect}]. \tag{5}$$

Thus, $\Delta > 0$ indicates that correct searches are more concentrated than incorrect ones. We use these quantities as diagnostics of search structure rather than optimization objectives, since greater basin diversity does not necessarily imply higher accuracy.

## 4.2 Main Results

**Game of 24.** Table 1 shows that BASIN improves accuracy from 66.0% to 72.0% with `gpt-4o-mini`, and from 43.0% to 65.0% with `Qwen3-27B` (+22pp, $p$<0.01). The gains occur under the same search budget and exact symbolic verifier. BASIN also increases $N_{\text{eff}}$,

Table 1: Results on Game of 24. $^{**}p < 0.01$, $^{\dagger}p < 0.10$.

| Model | Method | Acc. | $n_{\text{basins}}$ | $N_{\text{eff}}$ |
|---|---|---|---|---|
| `gpt-4o-mini` | ToT | 0.660 | 27.39 | 26.65 |
| | +BASIN | **0.720**$^{\dagger}$ | **27.94** | **27.29** |
| `Qwen3-27B` | ToT | 0.430 | 25.64 | 24.88 |
| | +BASIN | **0.650**$^{**}$ | **28.15** | **27.38** |

Table 3: Results on MuSR ($n$=300, nine rounds, $\lambda$=3.0). Sel. Eff. indicates selection efficiency.

| Model | Method | Acc. | Pass@k | Sel. Eff. | $n_{\text{basins}}$ | $N_{\text{eff}}$ |
|---|---|---|---|---|---|---|
| gpt-oss-120b | ToT | 0.633 | 0.863 | 0.734 | **8.59** | **6.86** |
| | +BASIN | **0.670**$^{*}$ | **0.903** | **0.742** | 8.58 | 6.85 |
| gpt-4o-mini | ToT | 0.607 | **0.857** | 0.708 | 6.76 | 5.07 |
| | +BASIN | **0.620**$^{\dagger}$ | 0.833 | **0.744** | **6.84** | **5.28** |

particularly for Qwen3-27B, suggesting
that ToT spends substantial compute revisiting structurally redundant arithmetic states. We observe similar findings for the BBH logical deduction task (Appendix C) with +13pp in accuracy.

**MuSR.** Table 3 reports results on 300 MuSR problems. The NLI-based semantic construction is inherently noisier than the exact structural basin definition used for Game of 24, which may limit how precisely the revisit penalty distinguishes genuinely different reasoning strategies. Unlike Game of 24, global diversity changes on MuSR are small. For gpt-oss-120b, BASIN improves both accuracy and Pass@k; for gpt-4o-mini, Pass@k decreases slightly while accuracy increases. We define selection efficiency as $\mathrm{Acc.}/\mathrm{Pass@}k$; BASIN achieves the highest selection efficiency for both models. This suggests that BASIN performance depends not only on exploration but also on the quality of the semantic basin representation. Appendix G shows that accuracy remains stable across alternative semantic constructions. Overall, the gains arise from reallocating search across strategies rather than simply maximizing basin coverage.

**Quality-aware selection.** We further investigate the exploration–exploitation trade-off in BASIN. Table 2 compares standard ToT, BASIN, and QA-BASIN using gpt-4 [Achiam et al., 2023]. On Game of 24, standard ToT obtains 67.0% accuracy, flat BASIN falls to 61.0%, and QA-BASIN reaches 70.0%. On MuSR, the corresponding accuracies are 52.0%, 58.3%, and 58.7%. Notably, on Game of 24 the three methods achieve nearly identical effective basin counts despite substantially different accuracies. This reinforces that the objective is not to maximize basin diversity itself, but to avoid redundant exploration without suppressing promising reasoning regions. QA-BASIN directly addresses this trade-off by weakening the revisit penalty for basins with stronger quality signals.

### 4.3 Generalization Across Tasks, Models, and Search

We next test whether basin-aware selection generalizes beyond the primary Game of 24 and MuSR settings. Table 4 reports accuracy under matched inference budgets on HumanEval and GSM-Hard across four models. Across these settings, QA-BASIN is generally the most robust formulation: it matches or improves upon standard ToT for all four models on HumanEval and achieves the best or tied-best accuracy in three of four GSM-Hard settings. In contrast, flat BASIN sometimes increases exploration without improving accuracy, consistent with an exploration–exploitation trade-off rather than a simple benefit from greater basin coverage. This pattern also extends to additional Game of 24 models: both basin-aware variants improve over ToT on Qwen2.5-7B-Instruct, while QA-BASIN improves Llama-3.3-70B-Instruct from 70% to 75% (Appendix B).

Table 2: QA-BASIN vs. baselines on gpt-4.

| Task | Method | Acc. | $n_{\text{basins}}$ | $N_{\text{eff}}$ |
|---|---|---|---|---|
| Game24 | ToT | 0.670 | 28.17 | 27.65 |
| | BASIN | 0.610 | 28.24 | 27.72 |
| | QA-BASIN | **0.700** | **28.25** | **27.73** |
| MuSR | ToT | 0.520 | 5.86 | 4.67 |
| | BASIN | 0.583 | **7.39** | **5.72** |
| | QA-BASIN | **0.587** | 7.32 | 5.64 |

**Generalization beyond ToT.** Because BASIN modifies candidate selection rather than the topology of a particular search algorithm, we also evaluate it with UCT-based Monte Carlo Tree Search (MCTS) on Game of 24 under matched search budgets. We use 50 simulations per problem and add the basin term only to UCT child selection (Table 5).

Table 5: Game of 24 with UCT-based MCTS.

| Model | MCTS | BASIN | QA-BASIN |
|---|---|---|---|
| gpt-4o-mini | .460 | .420 | **.640** |
| gpt-oss-120b | .240 | .270 | **.390** |
| Qwen2.5-7B-Instruct | .490 | .360 | **.550** |
| Llama-3.3-70B-Instruct | .610 | .650 | **.720** |

Table 4: **Generalization across tasks and models.** Accuracy under matched inference budgets.

| Task | Model | ToT | BASIN | QA-BASIN |
|---|---|---|---|---|
| HumanEval | gpt-4o-mini | .799 | .811 | **.817** |
| | gpt-oss-120b | .756 | .750 | **.793** |
| | Qwen2.5-7B-Instruct | .756 | .750 | **.780** |
| | Llama-3.3-70B-Instruct | **.817** | .799 | **.817** |
| GSM-Hard | gpt-4o-mini | .510 | **.530** | **.530** |
| | gpt-oss-120b | **.610** | **.610** | **.610** |
| | Qwen2.5-7B-Instruct | .440 | .450 | **.460** |
| | Llama-3.3-70B-Instruct | .440 | **.470** | .440 |

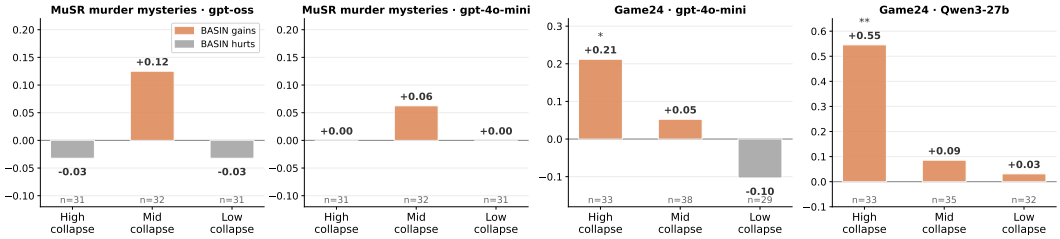


Figure 2: BASIN accuracy gain stratified by ToT basin collapse. Problems are grouped into tertiles by ToT $N_{\text{eff}}$. $^{**}p < 0.01$, $^{*}p < 0.05$ (one-sided McNemar test).

QA-BASIN improves over standard MCTS for all four models, by $+18$, $+15$, $+6$, and $+11$ pp, respectively. Flat BASIN, however, helps two models and hurts two. Since MCTS already contains an explicit exploration term, an additional unconditional revisit penalty can over-explore and displace promising regions. The quality-aware variant instead preserves high-quality basins while discouraging repeated visits to weaker ones. We observe the same transfer beyond tree search with Graph of Thoughts (GoT) on MuSR: BASIN improves accuracy from $57\%$ to $60\%$, while QA-BASIN further improves it to $64\%$. Notably, QA-BASIN achieves this gain with lower effective basin coverage than flat BASIN, again illustrating that effective basin-aware search requires balancing exploration with preservation of promising reasoning regions. Full results are reported in Appendix D.

Taken together, these results show that basin-aware selection transfers across tasks, model families, and search algorithms. They also motivate QA-BASIN as the preferred formulation when a reliable quality signal is available: it retains the benefit of escaping repeatedly explored low-quality basins while reducing the over-exploration that can arise from flat BASIN.

### 4.4 Collapse-Stratified Analysis

If reasoning basin collapse is an important failure mode, BASIN should help most when standard ToT repeatedly concentrates on a small set of strategies. We test this by splitting problems into tertiles according to standard-ToT $N_{\text{eff}}$: high-, mid-, and low-collapse. We then report paired accuracy differences $\Delta\text{acc} = \text{acc}_{\text{BASIN}} - \text{acc}_{\text{ToT}}$ in Figure 2. For MuSR, we use the murder-mystery subset to avoid mixing heterogeneous subtasks; Appendix A reports the remaining subsets.

The expected pattern is clearest on Game of 24. With `gpt-4o-mini`, BASIN gains $+0.21$ accuracy in the high-collapse group, compared with $+0.05$ in the mid-collapse group and $-0.10$ in the low-collapse group. `Qwen3-27B` shows the same qualitative trend, with most of the improvement concentrated among high-collapse problems.

MuSR is less monotonic: the largest gains occur in the mid-collapse group. Semantic $N_{\text{eff}}$ captures the amount of concentration but not whether the dominant reasoning basin is useful or misleading. This weaker alignment between collapse severity and the need for exploration may partly explain the smaller gains on MuSR relative to Game of 24. Overall, collapse severity is informative but insufficient by itself to determine when additional exploration will help.

### 4.5 Understanding Search Through the Redundancy Gap

The flat BASIN penalty is quality-agnostic: it depends on basin visitation rather than correctness. We use the redundancy gap $\Delta$ from Eq. (5) to analyze how the resulting concentration differs between successful and unsuccessful searches.

Figure 3 shows that standard ToT (Std) typically operates near $\Delta \approx 0$ or slightly below: correct and incorrect searches exhibit similar concentration patterns. BASIN consistently shifts $\Delta$ positive, concentrating successful searches around strong regions while dispersing repeated exploration among unsuccessful ones. This separation helps explain why a quality-agnostic revisit penalty can improve accuracy: its effect depends not simply on increasing diversity, but on restructuring where redundancy occurs.

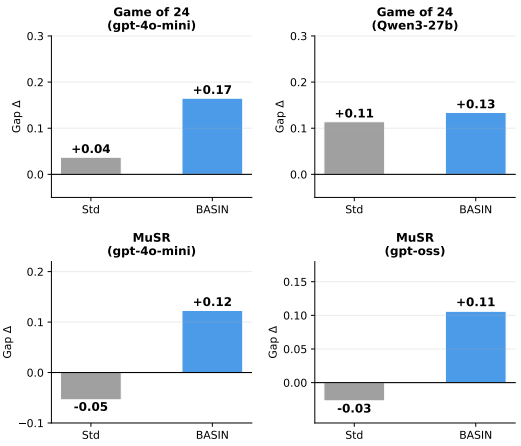


Figure 3: Redundancy gap $\Delta$ for ToT and BASIN.

The relationship is useful but not universal. For `Qwen3-27b` on Game of 24, ToT already has $\Delta = +0.11$, yet BASIN improves accuracy by +22pp. Since ToT solves only 43% of problems in this setting, substantial per-problem collapse onto incorrect arithmetic basins can remain even when the dataset-level gap appears favorable. Thus, $\Delta$ summarizes average search behavior but can obscure substantial per-problem heterogeneity.

The redundancy gap therefore provides a useful summary of how search concentration differs between successful and unsuccessful trajectories, but it does not by itself determine when additional exploration will improve accuracy. We therefore evaluate whether $\Delta$ can be used as a routing signal for choosing between standard search and BASIN. Across six model–task settings spanning Game of 24 and BBH, $\Delta$ alone selects the empirically better fixed policy in only $2/6$ cases. Combining it with a per-problem search-effort signal—the number of tree nodes explored before termination—increases this to $5/6$; Appendix H provides the full routing analysis. Thus, we treat the redundancy gap primarily as a diagnostic of search structure, while adaptive policies should combine it with additional search-state or quality signals.

### 4.6 Ablation Studies

**Effect of compute budget.** Figure 4 varies the number of reasoning rounds on 100 randomly selected MuSR problems with `gpt-oss-120b`, holding beam size fixed. Accuracy generally improves with additional reasoning depth, while $N_{\text{eff}}$ also increases. Basin-aware exploration is therefore most useful when the search budget is large for alternative strategies to develop into complete solutions.

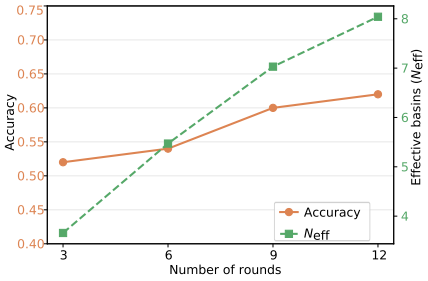


Figure 4: Effect of compute budget.

**Effect of penalty strength.** Figure 5 varies $\lambda \in \{0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0\}$ in the same MuSR setting. Accuracy peaks at $\lambda$=3.0 and is relatively stable over moderate values. $N_{\text{eff}}$ increases with $\lambda$, but accuracy is non-monotonic: weak penalties have little effect, whereas overly strong penalties can override useful base-score differences. This again shows that the goal is not maximal diversity, but an effective exploration–exploitation balance.

**How important is the quality signal?** Because QA-BASIN uses basin quality to modulate the revisit penalty, its performance depends on the informativeness of that signal. On MuSR with `gpt-4`, using an LLM-based quality signal yields 58.7% accuracy, compared with 58.3% for flat BASIN and 52.0% for standard ToT. Replacing this signal with the search heuristic reduces accuracy to 33.7%. Thus, quality-aware modulation is beneficial when the quality estimate is informative, but can be

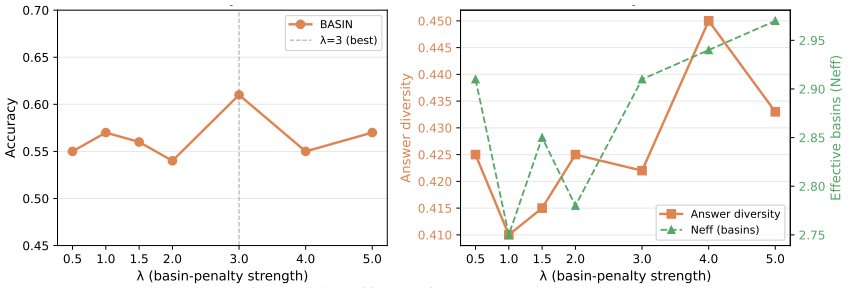

Figure 5: Effect of penalty strength.

actively harmful when it is poorly calibrated to correctness. We therefore recommend QA-BASIN when a meaningful quality signal is available and flat BASIN otherwise. Appendix E provides the full results and analyzes the discriminative quality of the heuristic score.

**Is BASIN Just Promoting Diversity?** The improvement from BASIN is not explained by generic diversity promotion. On Game of 24 with `gpt-4o-mini`, a Diverse Beam Search (DBS)-style baseline [Vijayakumar et al., 2016] achieves 64.0% accuracy, compared with 66.0% for standard ToT and 72.0% for BASIN. Thus, encouraging diverse candidates alone does not reproduce the gain from explicitly modeling strategy-level redundancy. We further vary the sampling temperature as an alternative way to increase token-level diversity (Fig. 6). Across $T \in \{0.7, 1.0, 1.2, 1.5\}$, the best higher-temperature ToT configuration reaches 68.0% accuracy, still below BASIN's 72.0% at $T = 0.7$. Increasing temperature within BASIN also does not improve performance. These results show that token-level or candidate-level diversity is not interchangeable with history-biased, basin-level selection: BASIN benefits from identifying and penalizing repeated reasoning strategies rather than simply making individual candidates more different.

### 4.7 Case Study

We illustrate how BASIN reallocates search effort on Game of 24; a MuSR example appears in Appendix I.

**Game of 24:** $1, 11, 11, 13$**.** Under standard ToT, the first-step beam contains the same symbolic state twice:

```
11 - 1 = 10,  13 - 11
= 2,  11 * 1 = 11,  11 + 11 = 22,  11 - 1 = 10.
```

The first and final candidates both map to basin `sub|10,11,13`, so one beam slot is spent revisiting the same arithmetic state. After its first visit, BASIN lowers the score of this basin, allowing the alternative $1+11 = 12$ to survive. This new basin leads to

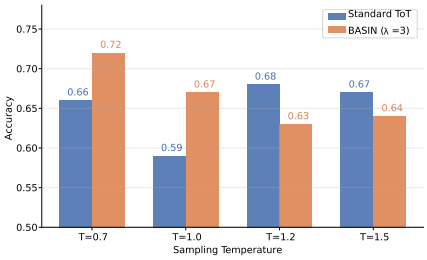

Figure 6: Effect of temperature $T$.

$$1 + 11 = 12 \longrightarrow 13 - 11 = 2 \longrightarrow 2 \times 12 = 24.$$

No new generator, operation, or verifier is introduced; the search simply allocates its existing beam budget across distinct symbolic strategies.

## 5 Discussion

**Why history-biased basin search can improve reasoning.** BASIN is motivated by the observation that inference-time reasoning search can over-commit to a small number of plausible but incomplete reasoning directions. In ToT-style search, early selections shape later expansions: if the beam repeatedly selects variants of the same hypothesis, subsequent rounds tend to elaborate that hypothesis rather than test alternatives. By tracking which reasoning basins have already been explored and penalizing repeated visits, BASIN introduces a history-dependent bias that reallocates search effort toward underexplored strategies and reduces the risk that all active states inherit the same error mode.

This mechanism is not equivalent to maximizing diversity. The Diverse Beam Search and temperature comparisons in §4.6 show that generic candidate- or token-level diversity does not reproduce the gains from modeling strategy-level redundancy, and increasing $N_{\text{eff}}$ can coincide with unchanged or

lower accuracy. The objective is therefore not maximal diversity, but avoiding excessive reuse of the same underlying reasoning strategy.

The redundancy gap $\Delta$ (§4.5) provides a diagnostic of this behavior. Standard ToT typically operates near $\Delta \approx 0$, whereas BASIN often shifts the gap positive. However, $\Delta$ alone selects the empirically better fixed policy in only $2/6$ settings; combining it with a search-effort signal succeeds in $5/6$ settings (Appendix H). We therefore treat $\Delta$ as a diagnostic of harmful concentration rather than a standalone routing criterion.

**Exploration versus exploitation.** Flat BASIN is most useful when baseline search repeatedly commits compute to an unproductive basin, but it can hurt when search has already identified a promising one. This effect is especially visible with stronger models and under MCTS, where an existing exploration mechanism can compound the additional history-based exploration pressure. The collapse-stratified results similarly show that additional exploration is most useful when baseline search is sufficiently concentrated. QA-BASIN addresses this trade-off by weakening the revisit penalty for basins with stronger quality evidence. Across tasks and models, it is more robust than the unconditional penalty, and under MCTS it improves over standard search for all four evaluated models. We therefore view QA-BASIN as the preferred formulation when a meaningful quality signal is available, with flat BASIN providing a simpler alternative when it is not.

**The importance of basin structure.** The effectiveness of history-biased search depends on the basin representation capturing meaningful reasoning redundancy. For symbolic tasks, basin membership can often be defined exactly; for open-ended reasoning, it must be approximated. On MuSR, SBERT-based similarity collapses trajectories into nearly a single basin ($N_{\text{eff}} \approx 1.04$), whereas NLI-based clustering over extracted hypotheses yields a richer strategy-level structure ($N_{\text{eff}} \approx 6.85$; Appendix F). At the same time, performance is robust within this semantic construction family: accuracy remains above standard ToT across tested entailment thresholds, NLI models, and hypothesis extractors, although basin counts and Pass@k vary more (Appendix G).

**Generality and limitations.** Results across HumanEval, GSM-Hard, Game of 24, MuSR, MCTS, and GoT suggest that reasoning-basin structure is not tied to a single task, model family, or search topology. More broadly, BASIN suggests *history-biased reasoning search* as a general principle: represent strategy-level redundancy through reasoning basins, then use accumulated search history to influence where inference compute is allocated.

Several limitations remain. Basin definitions are task-dependent, and semantic tasks require approximate representations with additional extraction and NLI cost. Neither basin coverage nor $N_{\text{eff}}$ determines whether the explored strategy is correct, while QA-BASIN additionally depends on the reliability of its quality signal. Our experiments also use matched inference budgets rather than characterizing the full accuracy–token-cost Pareto frontier. Finally, the redundancy gap is insufficient by itself for deciding when exploration is beneficial; adaptive policies combining basin visitation, quality estimates, and online search-state signals are a promising direction for future work.

## Acknowledgments

This work is supported by the National Science Foundation (NSF) Grant #2312862, NSF-Simons SkAI Institute, NSF CAREER #2440542, NSF #2533996, NSF #2621883, National Institutes of Health (NIH) #R01AG091762, NSF ACCESS Computing Resources, NAIRR, NRP, a Google Research Scholar Award, and Cisco gift grant.

## References

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc 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*, 2023. URL `https://openreview.net/forum?id=1PL1NIMMrw`.

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*, volume 36, pages 11809–11822, 2023.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8179, 2023.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62138–62160. PMLR, 2024. URL `https://proceedings.mlr.press/v235/zhou24r.html`.

Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zengmao Wang, Ziwei Liu, Bo Du, Xianglong Liu, and Dacheng Tao. Dynamic parallel tree search for efficient LLM reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 11233–11252, 2025.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=jenyYQzue1`.

Sai Shridhar Balamurali and Lu Cheng. Revisiting nli: Towards cost-effective and human-aligned metrics for evaluating llms in question answering. *arXiv preprint arXiv:2511.07659*, 2025.

Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.

Pratyush Tiwary and Michele Parrinello. From metadynamics to dynamics. *Physical review letters*, 111(23):230602, 2013.

Yihang Wang, Joao Marcelo Lamim Ribeiro, and Pratyush Tiwary. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Current opinion in structural biology*, 61:139–145, 2020.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html`.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=WZH7099tgfM`.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=5NTt8GFjUHkr`.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488, 2022. URL `https://openreview.net/forum?id=_3ELRdg2sgI`.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=yf1icZHC-l9`.

Weihao Liu, Dehai Min, and Lu Cheng. Latent thoughts tuning: Bridging context and reasoning with fused information in latent tokens. In *ICML*, 2026.

Yang Li. Policy guided tree search for enhanced LLM reasoning. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=NNWSNy4YB4. Poster.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Xiaofan Zhou, Huy Nguyen, Bo Yu, Chenxi Liu, and Lu Cheng. Adaptive stopping for multi-turn llm reasoning. *arXiv preprint arXiv:2604.01413*, 2026.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. CP-Router: An uncertainty-aware router between LLM and LRM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Xiaofan Zhou and Lu Cheng. What should the reflector see? an empirical study of evidence in reflective prompt optimization. *arXiv preprint arXiv:2609.32452*, 2026a.

Xiaofan Zhou and Lu Cheng. Streamlined reflective evolution for task-adaptive self-refinement pipelines. *arXiv preprint arXiv:2609.32458*, 2026b.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652, 2023.

Liangjie Huang, Dawei Li, Huan Liu, and Lu Cheng. Beyond accuracy: The role of calibration in self-improving large language models. In *2025 IEEE International Conference on Big Data (BigData)*, pages 2863–2872. IEEE, 2025.

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

Guoming Ling, Zhongzhan Huang, Yupei Lin, Junxin Li, Shanshan Zhong, Hefeng Wu, and Liang Lin. Neural chain-of-thought search: Searching the optimal reasoning path to enhance large language models. *arXiv preprint arXiv:2601.11340*, 2026. URL https://arxiv.org/abs/2601.11340.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016. URL https://arxiv.org/abs/1610.02424.

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. Diversity of thought improves reasoning abilities of large language models. *arXiv preprint arXiv:2310.07088*, 2023. URL https://arxiv.org/abs/2310.07088.

Vijay Lingam, Behrooz Omidvar Tehrani, Sujay Sanghavi, Gaurav Gupta, Sayan Ghosh, Linbo Liu, Jun Huan, and Anoop Deoras. Enhancing language model agents using diversity of thoughts. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa,

Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. *arXiv preprint arXiv:2211.10435*, 2022.

Aarohi Srivastava, Abhishek Rastogi, Abhinav Rao, and et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

OpenAI. GPT-4o system card, 2024. URL `https://arxiv.org/abs/2410.21276`.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL `https://arxiv.org/abs/2505.09388`.

OpenAI, Sandhini Agarwal, et al. gpt-oss-120b & gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

## A MuSR Per-Subtask Results

MuSR [Sprague et al., 2024] contains three reasoning subtasks: murder mysteries, team allocation, and object placement. Table 6 reports results across all three subtasks and both models. These results provide a finer-grained view of when a quality-agnostic basin-revisit penalty helps and when additional exploration can instead disrupt a promising reasoning region.

Table 6: BASIN vs. ToT on all three MuSR subtasks. $\Delta_{Acc}$ denotes the BASIN minus ToT accuracy difference. $N_{\text{eff}}$ is the mean effective basin count under standard ToT and $n$ is the sample size. Pass@$k$ is the fraction of problems for which at least one final candidate is correct.

| Model | Task | $n$ | ToT | BASIN | $\Delta_{Acc}$ | Pass@$k$ (ToT / BASIN) | $N_{\text{eff}}$ |
|---|---|---|---|---|---|---|---|
| `gpt-oss-120b` | Murder | 94 | 0.628 | **0.649** | +0.021 | 0.915 / **0.957** | 8.06 |
| | Team | 96 | **0.719** | **0.719** | +0.000 | 0.938 / **0.958** | 8.76 |
| | Object | 110 | 0.564 | **0.645** | +0.082 | 0.755 / **0.809** | 4.18 |
| `gpt-4o-mini` | Murder | 94 | 0.649 | **0.670** | +0.021 | **0.883** / 0.862 | 6.39 |
| | Team | 96 | 0.562 | **0.625** | +0.062 | **0.865** / 0.812 | 5.18 |
| | Object | 110 | **0.609** | 0.573 | −0.036 | **0.827** / **0.827** | 3.84 |

The results reinforce that basin concentration alone does not determine whether additional exploration will help. Low $N_{\text{eff}}$ can indicate harmful collapse onto an incorrect strategy, but it can also reflect useful agreement around a strong solution. Pass@$k$, final accuracy, and the nature of the dominant basin are therefore important for interpreting the effect of BASIN.

**Murder mysteries.** Both models show a small positive accuracy gain (`gpt-oss-120b`: $+0.021$, 9 wins vs. 7 losses; `gpt-4o-mini`: $+0.021$, 7 wins vs. 5 losses), although neither difference is statistically significant ($p = 0.40$ and $p = 0.39$, respectively). The collapse-stratified analysis in §4.4 further shows that the effect is heterogeneous across problems rather than uniformly determined by aggregate basin diversity. For `gpt-oss-120b`, standard ToT already achieves Pass@$k = 0.915$, leaving limited room to improve candidate coverage. For `gpt-4o-mini`, Pass@$k = 0.883$ similarly indicates that a substantial part of the remaining error comes from selecting among already discovered candidates rather than from failing to explore the correct answer entirely.

**Object placement.** The two models behave differently despite both exhibiting relatively low effective basin counts. For `gpt-oss-120b`, standard ToT has $N_{\text{eff}} = 4.18$ and Pass@$k = 0.755$, so nearly one quarter of problems contain no correct final candidate. BASIN improves accuracy by $+0.082$ ($p = 0.032$, 14 wins vs. 5 losses), consistent with additional strategy-level exploration recovering answers that standard search misses.

For `gpt-4o-mini`, standard ToT has an even lower $N_{\text{eff}} = 3.84$, yet Pass@$k$ is already $0.827$ and BASIN reduces final accuracy by $0.036$. Thus, low effective basin count is not by itself evidence of harmful collapse. In this setting, additional exploration can displace a useful consensus rather than recover a missing reasoning strategy.

**Team allocation.** `gpt-4o-mini` gains $+0.062$ accuracy ($p = 0.035$), whereas `gpt-oss-120b` is unchanged. The latter already exhibits high effective basin coverage ($N_{\text{eff}} = 8.76$), while `gpt-4o-mini` operates at $N_{\text{eff}} = 5.18$. The result is consistent with BASIN being most useful when the baseline search underexplores materially different strategies, but the object placement results above show why diversity statistics alone are insufficient to identify that regime.

**Summary.** Across the six task–model combinations, the two statistically significant positive results occur in settings where additional strategy-level exploration can recover or preserve useful alternatives. The negative result demonstrates the complementary failure mode: an unconditional revisit penalty can over-explore when concentration reflects useful exploitation rather than pathological collapse. This motivates the quality-aware formulation in Eq. (4), which weakens the penalty for high-quality basins.

## B Additional Game of 24 Models

To complement the main Game of 24 results, we evaluate Qwen2.5-7B-Instruct and Llama-3.3-70B-Instruct under the same matched-budget protocol. Table 7 reports accuracy for standard ToT, BASIN, and QA-BASIN.

Table 7: **Additional Game of 24 model results.** Accuracy under matched inference budgets. Bold denotes the best result within each model.

| Model | ToT | BASIN | QA-BASIN |
|---|---|---|---|
| Qwen2.5-7B-Instruct | 0.600 | **0.650** | 0.640 |
| Llama-3.3-70B-Instruct | 0.700 | 0.680 | **0.750** |

The additional models exhibit the same exploration–exploitation trade-off observed elsewhere. Flat BASIN improves Qwen2.5-7B-Instruct from $60\%$ to $65\%$, but slightly reduces accuracy for Llama-3.3-70B-Instruct from $70\%$ to $68\%$. In contrast, QA-BASIN reaches $75\%$ on Llama-3.3-70B-Instruct, illustrating the benefit of preserving promising basins when unconditional exploration can displace high-quality trajectories.

## C BBH Logical Deduction

To evaluate transfer beyond the main benchmarks, we apply BASIN to the *Logical Deduction* subtask of BIG-Bench Hard [Srivastava et al., 2022]. The task requires ordering objects from relational constraints and therefore differs from both arithmetic search and open-ended abductive reasoning. We evaluate $n = 100$ problems with `gpt-4o-mini` and $\lambda = 3.0$.

Table 8: Results on BBH Logical Deduction (`gpt-4o-mini`, $n$=100, $\lambda$=3.0). $^*p < 0.05$ using a one-sided McNemar test.

| Method | Acc. | $n_{\text{basins}}$ | $N_{\text{eff}}$ |
|---|---|---|---|
| Standard ToT | 0.400 | 6.49 | 4.50 |
| ToT + BASIN | **0.530**$^*$ | **7.11** | **5.60** |

BASIN improves accuracy by 13pp over standard ToT ($p = 0.018$, 23 wins vs. 10 losses) while increasing $N_{\text{eff}}$ from 4.50 to 5.60. This result provides an additional example in which reallocating search toward distinct reasoning strategies improves accuracy. As elsewhere in the paper, however, the increase in $N_{\text{eff}}$ should be interpreted as a description of the changed search behavior rather than as the objective itself.

## D Graph-of-Thought Backbone

We additionally evaluate BASIN and QA-BASIN with a Graph-of-Thought (GoT) backbone [Besta et al., 2024] on MuSR (`gpt-oss-120b`, $n$=100, 12 calls per problem). GoT extends ToT with an explicit aggregation step that merges top-scoring trajectories into a refined answer. We apply the basin-revisit penalty during search selection while leaving the aggregation mechanism unchanged.

Table 9: GoT results on MuSR (`gpt-oss-120b`, $n$=100). Agg-$\Delta$ is the fraction of problems on which GoT aggregation changes the beam answer; Agg-Correct is the fraction of changed answers that are correct.

| Method | Acc. | $N_{\text{eff}}$ | Escape% | Agg-$\Delta$% | Agg-Correct% |
|---|---|---|---|---|---|
| GoT | 0.570 | 5.35 | 48.3 | 17 | 64 |
| GoT + BASIN | 0.600 | **5.69** | **51.6** | **18** | **69** |
| GoT + QA-BASIN | **0.640** | 5.12 | 45.8 | 16 | 68 |

Flat BASIN improves GoT accuracy by 3pp, from 57% to 60%, while increasing $N_{\text{eff}}$ from 5.35 to 5.69. QA-BASIN further improves accuracy to 64%, a 7pp gain over standard GoT, despite reducing $N_{\text{eff}}$ to 5.12 and the escape rate to 45.8%. This again shows that improved reasoning does not require maximizing exploration: quality-aware selection can retain stronger reasoning regions while avoiding unproductive revisits.

The aggregation step changes the beam answer on 16–18% of problems and selects the correct answer in 64–69% of those cases. The mechanisms therefore act at complementary stages: basin-aware selection determines which strategies survive search, while GoT aggregation combines the resulting trajectories afterward. These results provide additional evidence that both the basin mechanism and its quality-aware extension transfer beyond a single tree-search controller.

## E Sensitivity to the Quality Signal

The quality-aware formulation in Eq. (4) assumes that the signal used to estimate basin quality is informative. We therefore examine both the quality of the underlying scoring signal and the sensitivity of QA-BASIN to different choices of that signal.

**Quality of heuristic scores.** We first evaluate whether the heuristic scoring function used during search reliably distinguishes correct from incorrect trajectories. Table 10 reports its discriminative performance on MuSR with `gpt-oss-120b`. The score AUC is close to chance for both GoT and GoT+BASIN (0.510 and 0.516, respectively), indicating that the heuristic provides little direct information about trajectory correctness. Nevertheless, BASIN improves final accuracy from 57.0and selection efficiency from 0.671 to 0.706. This suggests that the gain in this setting does not arise from a strong verifier, but from changing which reasoning strategies survive search.

**Effect on quality-aware selection.** A weak quality signal is more consequential for QA-BASIN, because the signal directly controls the strength of the basin-revisit penalty. Table 11 compares

Table 10: Heuristic scoring quality on MuSR (`gpt-oss-120b`, $n$=100, six rounds). AUC is computed over terminal trajectories.

| Method | Acc. | Pass@$k$ | Sel. Eff. | Score AUC | Agg. changed |
|---|---|---|---|---|---|
| GoT | 0.570 | 0.850 | 0.671 | 0.510 | 17% |
| GoT + BASIN | **0.600** | 0.850 | **0.706** | **0.516** | 18% |

QA-BASIN using the heuristic score with an LLM-based quality signal on MuSR with `gpt-4`. Using the weak heuristic substantially reduces accuracy to 33.7standard ToT (52.0`gpt-4o-mini` as the quality signal yields 58.7over flat BASIN and substantially outperforming standard ToT.

Table 11: Effect of quality-signal choice on QA-BASIN for MuSR (`gpt-4`, $\lambda$=0.5, $n$=300).

| Method | Quality signal | Acc. | $n_{\text{basins}}$ | $N_{\text{eff}}$ |
|---|---|---|---|---|
| Standard ToT | – | 0.520 | 5.86 | 4.67 |
| BASIN | – | 0.583 | 7.39 | 5.72 |
| QA-BASIN | Heuristic | 0.337 | 3.81 | 3.36 |
| QA-BASIN | LLM (`gpt-4o-mini`) | **0.587** | 7.32 | 5.64 |

These results clarify the roles of the two formulations. Flat BASIN does not require an estimate of basin quality and can therefore be used when no reliable quality signal is available. QA-BASIN provides a better exploration–exploitation mechanism when an informative quality signal is available, because it can preserve promising basins while continuing to penalize repeated visits to lower-quality ones. However, a poor quality signal can incorrectly protect weak basins or suppress useful exploration, as illustrated by the heuristic result above. We therefore view QA-BASIN as the preferred formulation when a meaningful verifier or quality estimate is available, rather than as uniformly superior independent of signal quality.

## F Basin Definition: NLI versus Embedding Similarity

For open-ended tasks such as MuSR, reasoning basins require an approximate semantic equivalence relation. We use NLI-based clustering because direct embedding similarity produces overly coarse partitions of the reasoning space. Table 12 compares the two representations on a MuSR diagnostic subset.

Table 12: Basin structure under SBERT cosine similarity and NLI-based semantic clustering on MuSR. Very small $N_{\text{eff}}$ indicates that most trajectories are assigned to the same basin.

| Basin definition | Model | Mean basins | $N_{\text{eff}}$ |
|---|---|---|---|
| SBERT cosine | `gpt-4o-mini` | 1.40 | 1.37 |
| SBERT cosine | `gpt-oss-120b` | 1.05 | 1.04 |
| NLI semantic | `gpt-oss-120b` | **8.58** | **6.85** |

SBERT cosine similarity merges most MuSR trajectories into one or two basins. Different reasoning traces for the same problem reuse substantial narrative context, so explanations supporting different hypotheses can remain close in embedding space. A representation with $N_{\text{eff}}$ near one provides little useful strategy-level structure because nearly every candidate is treated as belonging to the same region.

NLI-based clustering instead compares the propositional content of compact extracted hypotheses. Paraphrases supporting the same answer and argument can be grouped together, while incompatible hypotheses remain separate. This produces a substantially richer partition and better matches the type of strategy-level redundancy that BASIN is intended to track.

More generally, exact symbolic or structural keys are preferable when they are available. Semantic clustering is a fallback for tasks without a deterministic equivalence relation; the selection mechanism itself is agnostic to how basin membership is constructed.

## G Semantic Basin Sensitivity

Semantic basin construction introduces learned components and clustering choices. We therefore rerun the full search while varying the entailment threshold, NLI model, and hypothesis extractor rather than merely reclustering fixed trajectories post hoc. Table 13 reports results for MuSR with `gpt-4o-mini`; standard ToT achieves accuracy $0.607$ and Pass@$k = 0.857$.

Table 13: Sensitivity of BASIN to semantic basin construction on MuSR.

| **Basin definition** | **Acc.** | **Pass@$k$** | $n_{\text{basins}}$ | $\boldsymbol{N_{\text{eff}}}$ |
|---|---|---|---|---|
| Default ($\tau_e$=0.45, v3-small, default extractor) | 0.620 | 0.833 | 6.84 | 5.28 |
| $\tau_e$=0.30 | **0.640** | 0.780 | 3.78 | 3.17 |
| $\tau_e$=0.60 | **0.640** | 0.760 | 6.28 | **5.78** |
| Alternative NLI model (v3-base) | **0.640** | 0.780 | 4.36 | 3.74 |
| Alternative extractor (`gpt-4`) | 0.620 | 0.740 | 4.74 | 4.09 |

Accuracy remains between $0.620$ and $0.640$ and exceeds standard ToT under every tested semantic basin definition despite substantial changes in basin counts and $N_{\text{eff}}$. Pass@$k$ is more sensitive, particularly to the choice of hypothesis extractor. Thus, the semantic representation affects the detailed search trajectory, but the observed accuracy improvement is robust across the tested configurations.

The contradiction ceiling has no observable effect over $\tau_c \in \{0.20, 0.30, 0.40\}$ in these experiments, suggesting that the entailment criterion already removes most incompatible pairs. We nevertheless view semantic basin construction as a genuine source of modeling sensitivity and report these results to make that dependence explicit.

## H Redundancy Gap as a Routing Signal

The redundancy gap is informative about search behavior, but it is not sufficient by itself as a deployment rule. We test whether $\Delta_\rho$, computed from standard-search traces, predicts whether BASIN improves accuracy. We additionally measure $\Delta N_{\text{eff}}$, the change in effective basin count under BASIN, where larger values indicate a stronger increase in strategy-level exploration.

Following the rebuttal analysis, we define four regimes using the signs of $\Delta_\rho$ and $\Delta N_{\text{eff}}$, with a threshold of $0.3$ for the latter:

- **Explore-clear:** $\Delta N_{\text{eff}} > 0.3$ and $\Delta_\rho \leq 0$;
- **Exploit-clear:** $\Delta N_{\text{eff}} \leq 0.3$ and $\Delta_\rho > 0$;
- **Restructuring:** $\Delta N_{\text{eff}} \leq 0.3$ and $\Delta_\rho \leq 0$;
- **Ambiguous:** $\Delta N_{\text{eff}} > 0.3$ and $\Delta_\rho > 0$.

The aggregate rule selects BASIN for the entire experiment whenever $\Delta_\rho \leq 0$. The combined rule uses the same decision except in the Ambiguous regime, where it additionally uses a per-problem search-effort signal already available from the standard run: the number of explored tree nodes before termination, `n_nodes`. Problems at or below the experiment-specific median are routed to BASIN; those above the median remain under standard ToT.

Table 14: **Redundancy gap as a routing signal.** $\Delta_\rho$ is computed from standard-search traces and $\Delta N_{\text{eff}}$ is the change in effective basin count under BASIN. Acc. (routed) uses the combined per-problem routing rule in the Ambiguous regime. Agg. and Comb. indicate whether the aggregate and combined rules, respectively, select the empirically better fixed policy.

| **Experiment** | $\boldsymbol{\Delta_\rho}$ | $\boldsymbol{\Delta N_{\text{eff}}}$ | **Acc. (std)** | **Acc. (BASIN)** | **Acc. (routed)** | **Regime** | **Agg.** | **Comb.** |
|---|---|---|---|---|---|---|---|---|
| Game24 / `gpt-4o-mini` | +0.037 | +0.638 | 0.660 | 0.720 | **0.760** | Ambiguous | No | Yes |
| Game24 / `gpt-oss` | +0.486 | −0.184 | **0.380** | 0.370 | **0.380** | Exploit-clear | Yes | Yes |
| Game24 / `Qwen3-397B` | +0.230 | +0.935 | 0.380 | **0.490** | 0.480 | Ambiguous | No | Yes |
| Game24 / `Qwen3-27B` | +0.114 | +2.502 | 0.430 | 0.650 | **0.700** | Ambiguous | No | Yes |
| BBH / `gpt-oss` | −0.430 | +0.921 | 0.520 | **0.620** | **0.620** | Explore-clear | Yes | Yes |
| BBH / `gpt-4o-mini` | −0.349 | +0.356 | **0.740** | 0.700 | 0.700 | Explore-clear | No | No |

Using $\Delta_\rho$ alone selects the empirically better fixed policy in only $2/6$ settings ($33.3\%$). Its main failure mode is the Ambiguous regime: all three Ambiguous experiments have $\Delta_\rho > 0$, which would favor standard search under the aggregate rule, yet BASIN improves accuracy in all three. Incorporating the already-available `n_nodes` signal raises the routing decision accuracy to $5/6$ settings ($83.3\%$).

More importantly, per-problem routing improves over both globally fixed policies in two of the three Ambiguous settings. On Game24/`gpt-4o-mini`, routed accuracy reaches $0.760$, compared with $0.660$ for standard ToT and $0.720$ for BASIN. On Game24/`Qwen3-27B`, routing reaches $0.700$, compared with $0.430$ and $0.650$, respectively. In the remaining Ambiguous setting, routing remains close to the better fixed policy ($0.480$ vs. $0.490$).

These results reinforce the interpretation used in the main paper: the redundancy gap is a useful diagnostic of search structure, but it is insufficient as a standalone criterion for deciding when to increase exploration. Combining redundancy information with inexpensive problem-level search-state signals provides a more promising basis for adaptive structure-aware search.

# I Case Study: MuSR Murder Mystery: `murder_mysteries_185`

This MuSR example illustrates reasoning basin collapse in a semantic setting. The story concerns the death of Wilhelmina by crossbow. Two salient suspects are Isabelle and Nicole. Isabelle is a yoga instructor and member of an archery club who was present in the kitchen during the murder. Nicole owns an authentic medieval crossbow, remained at the crime scene throughout the day, and is associated with a pattern of suspicious deaths among acquaintances. The correct answer is **Nicole**.

Under standard ToT, the search predicts **Isabelle**. Although the surface forms of the generated explanations differ, most terminal hypotheses reuse the same core strategy: Isabelle had crossbow-related skill and was present at the scene. Representative hypotheses include:

> *"Isabelle is most likely the murderer because she is a skilled crossbow practitioner through her archery-club membership and was physically present in the kitchen at the time of Wilhelmina's death."*

> *"The most probable culprit is Isabelle—her confirmed crossbow expertise and attendance at the yoga session held in the victim's kitchen at the time of the killing place her squarely at the scene."*

> *"Isabelle committed the murder: she practises crossbow shooting regularly and her own account places her in Wilhelmina's kitchen during the window of the crime."*

These trajectories differ lexically but are equivalent at the strategy level: they predict the same answer and rely on the same central hypothesis (*crossbow skill* plus *presence at the scene*). They are therefore assigned to the same semantic reasoning basin. Repeated selection from this basin causes the search budget to elaborate the same explanation rather than testing materially different alternatives.

BASIN reduces the relative score of further revisits to the Isabelle-centered basin, allowing the search to retain a distinct Nicole-centered explanation:

> *"Nicole is the most likely murderer: she owns a genuine medieval crossbow displayed in her home, the victim was killed in Nicole's own kitchen during a visit Nicole hosted, and multiple people in Nicole's social circle have died under similarly mysterious circumstances."*

This trajectory belongs to a different semantic basin: it predicts *Nicole* and relies on a different explanatory strategy combining *weapon ownership*, *crime-scene ownership*, and a *pattern of suspicious deaths*. Preserving this alternative changes the composition of the final candidate set and allows the correct answer to be selected.

Together with the symbolic case study in the main paper, this example illustrates the common mechanism underlying reasoning basin collapse. The surface manifestation differs across domains, but in both cases search spends multiple selections on states that instantiate the same underlying strategy. BASIN uses this structure to discourage redundant revisits and reallocate inference budget toward underexplored alternatives.

## J Prompt Templates

All generation prompts are identical between standard search and BASIN; the methods differ only in the selection rule. MuSR additionally uses the same hypothesis-extraction and semantic-clustering pipeline for all compared search conditions. We list the principal prompts below.

### Game of 24 — Step Proposal

At each search depth, the model receives the current remaining numbers and proposes up to five candidate next steps. The system instruction and few-shot examples are fixed; only the final `Input:` line changes.

Game of 24 proposal prompt

```
[System] You are a Game of 24 expert.  At each step, you are given the CURRENT
remaining numbers.  Choose exactly two of them, apply one arithmetic operation
(+, -, *, /), and list possible next steps.  Format each step as:  A op B = C
(remaining:  X Y Z) where X Y Z are the numbers left after replacing A and B
with C. List up to 5 diverse candidate steps.

Input:  4 4 6 8 Possible next steps: 4 + 8 = 12 (remaining:  4 6 12) 6 - 4 = 2
(remaining:  2 4 8) 4 * 6 = 24 (remaining:  4 8) 4 * 8 = 32 (remaining:  4 6) 6
+ 8 = 14 (remaining:  4 4 14)

Input:  2 9 10 12 Possible next steps: 12 * 2 = 24 (remaining:  9 10) 2 + 12 =
14 (remaining:  9 10 14) 10 - 2 = 8 (remaining:  8 9 12) 10 - 9 = 1 (remaining:
1 2 12) 9 + 12 = 21 (remaining:  2 10 21)

[additional few-shot examples omitted for brevity]

Input:  remaining numbers Possible next steps:
```

### MuSR — Reasoning Generation

Each generation round produces one candidate reasoning trajectory. The system prompt is shared across rounds and search methods.

MuSR generation prompt

```
[System] You are a careful reasoning assistant.  Always end your response with
'Answer:  X' on its own line, where X is a single uppercase letter (A, B, C,
...).  Never skip the Answer line.

[User] full MuSR problem text and answer choices
```

### MuSR — Structured State Extractor

After each reasoning-generation call, a separate `gpt-4o-mini` call extracts a compact structured representation from the raw trace. The `main_hypothesis` field is used as the semantic representation for NLI-based basin construction; the remaining fields support downstream analysis and selection.

MuSR extractor prompt

```
You are given a reasoning trace that answers a multiple-choice question.
Extract the following information as a JSON object with EXACTLY these five keys:

  'final_answer''    :  the answer letter chosen (A / B / C / D / E)
'main_hypothesis' :  ONE sentence stating the single most important
argument for that answer   'key_evidence''   :  a JSON list of 2-3 short strings
citing specific
facts from the story
'eliminated_option':  ONE sentence naming which option was ruled out
```

```
and the decisive reason   ''reasoning_summary'':  ONE sentence describing the
overall reasoning                    strategy used

Return ONLY the JSON object -- no markdown fences, no explanation, nothing else.

Reasoning trace: ```
{trace}
```
```

## K Compute Resources

**LLM inference.** LLM inference is performed through external model APIs, so the experiments do not require local GPU inference. Experiments are parallelized across problems using multi-threaded API calls. The broader camera-ready evaluation includes the model families reported in the main tables, while the original MuSR and Game-of-24 experiments use `gpt-4o-mini`, `gpt-4`, `gpt-oss-120b`, and `Qwen3-27B`.

**Semantic representation cost.** For MuSR, both standard ToT and basin-aware search use the same semantic representation pipeline. Each problem uses 18 reasoning-generation calls and 18 hypothesis-extraction calls, for 36 LLM calls in total. Thus, hypothesis extraction adds 100identical across the compared MuSR search conditions. Consequently, the MuSR experiments isolate the effect of the selection rule conditional on using the same extraction and semantic-clustering machinery.

The local neural component is the NLI model used for semantic basin assignment, `cross-encoder/nli-deberta-v3-small`, which runs on CPU. A bidirectional pairwise comparison takes approximately 11.8,ms when batched at 32. Clustering roughly 18 states requires approximately 4–11,s per problem, plus a one-time 2.6,s model-loading cost per worker. The current implementation recomputes same-answer pairwise scores rather than updating the clusters incrementally, so these timings should be viewed as those of an unoptimized implementation.

For tasks with deterministic structural basin definitions, no hypothesis extraction or NLI computation is required for basin assignment. The additional search-side computation is limited to constructing the basin key, maintaining visit statistics, and modifying the candidate-selection score.

**Hardware.** Local experiments were run on a MacBook Pro with an Apple M2 8-core CPU and 16,GB unified memory. No local GPU is required for the reported basin tracking or semantic-clustering computations.

**API usage.** The experiments require substantially more API inference than a single generation baseline because inference-time search evaluates multiple reasoning trajectories. On MuSR, the semantic representation pipeline further doubles the number of LLM calls relative to generation alone, as described above. We therefore do not characterize semantic basin construction as computationally negligible; its cost is a limitation of the current open-ended implementation.
```

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All quantitative claims in the abstract and introduction are supported by experimental results in Experiments and Appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section Discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our method is empirical, does not include theoretical analysis.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All major hyperparameter and prompt templates are introduced in Experiments and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The anonymous supplement includes the code and data.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Experiments, Appendix, and Supplementary code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We included $^{*}p < 0.05$, $^{\dagger}p < 0.10$ (one-sided McNemar) for all main results in the main text.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work analyzes publicly available datasets and does not involve human subjects, personal data, or dual-use concerns.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: No negative societal impacts are foreseen for this basic research contribution.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The models and data in this paper (regime-switching time series analysis) pose no risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide all the citations that closely related to our work.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets or pre-trained models are released with this submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: No, not as part of the research methodology. LLMs were not used in the proposed method, experiments, data generation, evaluation, or analysis pipeline. Any LLM use was limited to writing and editing assistance, which the checklist excludes from mandatory declaration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.