When Should Language Models Search? A Topology-Aware Analysis of Reasoning Strategies

Dibyanayan Bandyopadhyay, MD Tanzeel Adam Khan, Asif Ekbal
Department of Computer Science and Engineering
Indian Institute of Technology Patna, India
{dibyanayan@gmail.com, tanzeelkhan78687@gmail.com, asif@iitp.ac.in}

Abstract

Search-based reasoning and chain-of-thought (CoT) prompting offer two contrasting inference-time paradigms for language models: CoT commits to a single linear trajectory, while search explores alternative partial reasoning paths before selecting an answer. In this work, we study this trade-off using A* as a controlled instantiation of heuristic-guided search. Across StrategyQA, LogicQA, HotpotQA, GSM8K, and MATH, and across three model scales, we find that search is not a universal upgrade over CoT; instead, the two strategies solve substantially different subsets of instances. To explain this complementarity, we introduce a *Reasoning Topology Framework* based on structural features computable before generation. A derived branching coefficient is strongly associated with when search helps across mixed-benchmark topology clusters, suggesting that the CoT–search boundary is partly explained by problem structure. We further train a topology-aware router that selects between CoT and A* without observing generated traces. Despite this restriction, it outperforms post-hoc routers based on semantic entropy and LLM-as-critic judgments. We attribute this to a *Fluency–Correctness Asymmetry*: fluent traces are often incorrect, making trace-level adjudication unreliable. Overall, our results suggest that adaptive reasoning should route by problem structure, deploying search only when the input topology warrants its higher cost. Code: https://github.com/newcodevelop/astar-cot-reasoning.

1 Introduction

Chain-of-thought (CoT) prompting (Wei et al., 2022) has become the standard way to elicit step-by-step reasoning from large language models (LLMs). By generating intermediate reasoning steps before a final answer, CoT often improves performance on arithmetic, commonsense, and symbolic tasks. However, its core generation process is still strictly *linear*: the model generates a single trace $s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_T$, and later steps are conditioned on all earlier generated contents. This design creates three potential problems. First, an early mistake can propagate through the entire trace because the model has no built-in backtracking mechanism. Second, questions that require integrating several independent pieces of evidence can be difficult to solve with a single left-to-right narrative. Third, CoT has no explicit search objective that trades off trace length and alternative hypotheses.

These limitations motivate search-based reasoning. If the model can explore multiple partial traces, then it can reconsider early commitments, compare candidate branches, and prioritize promising reasoning states. In this paper, we study a budgeted A*-search procedure over natural-language reasoning traces. The language model generates successor steps, a step cost penalizes both trace length and low-confidence steps, and a heuristic prioritizes which partial trace to expand next.

But search is not free of problems. Exploring multiple branches can also waste computation on dead-end decompositions or introduce noise on problems that a single well-chosen chain would solve directly. In our experiments, we find that A* occasionally *hurts* by overthinking simple questions (Section 3.5). The central question is therefore not "is search better?" but rather: *which structural properties of a reasoning problem predict whether branching will help?*

Concretely, we ask the following research questions: (i) Does guided search dominate CoT, or do they trade wins? (ii) When they disagree, is the split random or systematic? (iii) Can we predict which method will win from structural features of the input alone, before generating any reasoning trace from either CoT or the search procedure? (iv) Does such a pre-generation router actually outperform post-hoc adjudication (such as an LLM

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 4663–4681
November 6-10, 2026 ©2026 Association for Computational Linguistics

based judge) that sees full reasoning traces?

Motivating example. On StrategyQA, linear CoT can pursue a locally plausible but ultimately irrelevant chain, while A* can surface the decisive evidence earlier by exploring alternative partial traces. For example (Figure 1), on the question "Is shrimp scampi definitely free of plastic?", CoT misses the key fact of microplastic accumulation in seafood, whereas A* reaches that premise early and answers correctly. This illustrates the central hypothesis of the paper: CoT and search are complementary, and the real challenge is deciding when to use which. This pattern persists at scale. Across all three QA benchmarks, CoT and A* solve different subsets of examples. On StrategyQA, A* outperforms CoT (74.80% vs. 64.52%), yet the oracle reaches 84.59%. On LogicQA, CoT is slightly stronger (45.78% vs. 44.85%), but the oracle still reaches 59.29%. Together, these gaps show that the main opportunity is *strategy selection*: different questions favor different reasoning strategies.

Why A* specifically? Our goal is not to champion A* as the best search algorithm for LLMs, but to use it as a controlled diagnostic. A* separates the key ingredients of guided search; a priority frontier, a path cost, and a heuristic estimate, without entangling them with evaluator design (as in ToT), rollout policies (as in MCTS), or fixed-width pruning (as in beam search). This makes it easier to isolate *when* the act of branching itself is responsible for gains.

We do not claim that all search algorithms will yield the same gains or the same routing boundary. Problem structure may determine whether there is an opportunity for branching to help, while the particular search algorithm determines how effectively that opportunity is used. The quantitative boundary reported here is therefore specific to our budgeted A* implementation.

Contributions. Our contributions: (1) We compare CoT and A* across five benchmarks (StrategyQA, LogicQA, HotpotQA, GSM8K, MATH) and three model scales (0.5B, 8B, 14B), finding that neither uniformly dominates. (2) We quantify instance-level complementarity: CoT and A* solve genuinely different subsets, and the oracle gap is large enough to make routing worthwhile. (3) We introduce 13 pre-generation topology features and show that a single derived quantity, branching coefficient, predicts A* advantage better than dataset identity ($R^2 = 0.91$). (4) We show that a topology-aware router outperforms post-hoc trace evaluation, and diagnose why through a Fluency-Correctness Asymmetry analysis.

2 Methodology

We first formalize CoT and search-based reasoning under a common notation and then describe the routing mechanisms that choose between them.

2.1 Notation and Problem Setup

Let $\mathcal{Q} = (q, C, \mathcal{O})$ denote a reasoning problem, where q is the question, C is an optional context passage, and $\mathcal{O} = \{o_1, \ldots, o_K\}$ is the candidate answer set for multiple-choice tasks or the output space for free-form tasks. A *reasoning trace* $\tau = (s_1, s_2, \ldots, s_T)$ is an ordered sequence of natural-language reasoning steps, and a prediction function $\phi(\tau) \in \mathcal{O}$ extracts the final answer from the trace. Let $\mathcal{M}_\theta$ denote a language model with parameters θ.

2.2 Chain-of-Thought (CoT) Reasoning

In CoT prompting, the model generates a single reasoning trace autoregressively: $\tau^{\text{CoT}} = (s_1, s_2, \ldots, s_T)$ and $s_t \sim \mathcal{M}_\theta(\cdot \mid q, C, \mathcal{O}, s_{1:t-1})$. Generation terminates when the model emits a terminal answer pattern, and the final prediction is $\hat{y}^{\text{CoT}} = \phi(\tau^{\text{CoT}})$. CoT explores exactly one element of the trace space with no mechanism for backtracking or branch comparison.

2.3 A* Search over Reasoning Traces

We therefore search over the space of reasoning traces using an A*-style priority rule, but with a finite node budget and a multi-goal voting procedure for tractability.

State space. The search space is a tree $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ where each node $n \in \mathcal{S}$ represents a partial reasoning state: $n = (\tau_n, d_n)$, $\tau_n = (s_1, \ldots, s_{d_n})$. Here, τ_n is the partial trace stored at node n and $d_n = |\tau_n|$ is its depth. The root node n_0 has $\tau_{n_0} = \emptyset$ and $d_{n_0} = 0$. An edge $(n, n') \in \mathcal{E}$ exists when $\tau_{n'} = \tau_n \oplus s$ for some reasoning step s.

Successor generation. Given a node n with state (τ_n, d_n), we ask the language model to propose N_c candidate next steps:

$$\{(s^{(j)}, c^{(j)})\}_{j=1}^{N_c} = \text{GS}(\mathcal{M}_\theta, q, C, \mathcal{O}, \tau_n, N_c). \quad (1)$$

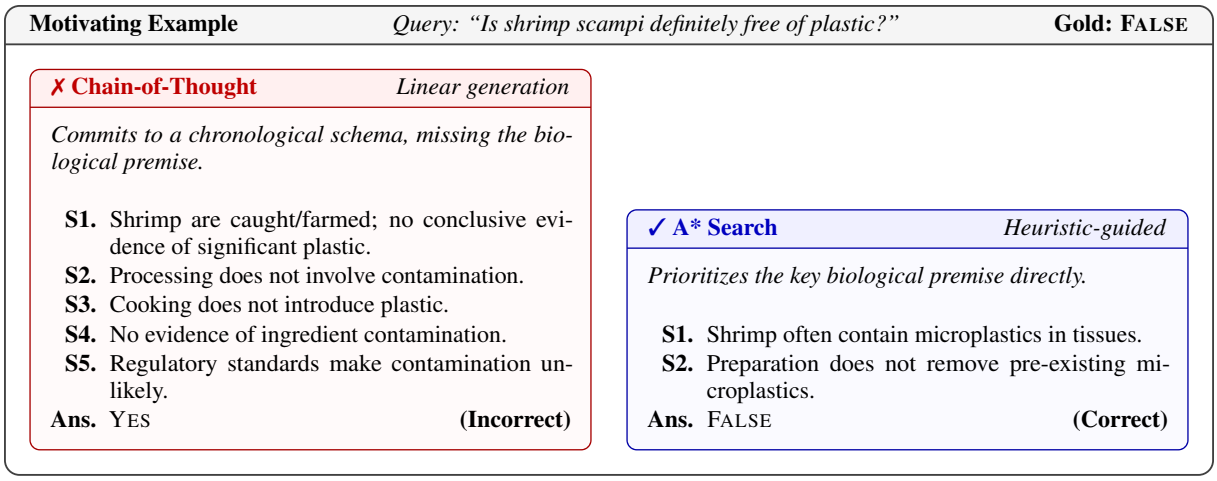


Figure 1: Comparison of reasoning trajectories on a StrategyQA example. CoT follows a plausible but misdirected chronological schema, while A* prioritizes the key biological fact, microplastic presence in shrimp, and reaches the correct conclusion with fewer, more relevant steps.

GS refers to the step-generation function. Each candidate consists of step text $s^{(j)}$ and a self-reported confidence $c^{(j)} \in [0.0, 1.0]$. The candidate induces a successor node n'_j with trace $\tau_{n'_j} = \tau_n \oplus s^{(j)}$ and depth $d_{n'_j} = d_n + 1$.

Goal condition. A node is a goal node if its last step contains a terminal answer pattern:

$$\textsc{IsGoal}(n) = \mathbb{1}\Big[\exists p \in \mathcal{P} : p \subset s_{d_n}\Big], \quad (2)$$

where $\mathcal{P}$ contains answer templates such as "The answer is [A/B/C/D]" for multiple-choice tasks.

Path cost. The cumulative cost of a node trades off reasoning length and uncertainty:

$$g(n) = \sum_{t=1}^{d_n} \big[1+(1-c_t)\big] = d_n + \sum_{t=1}^{d_n}(1-c_t). \quad (3)$$

Each step contributes a base cost of 1, and lower-confidence steps incur an additional penalty. This makes shorter, higher-confidence traces cheaper than longer or more uncertain ones.

We treat the self-reported confidence c_t only as a bounded search-ranking signal, not as a calibrated probability score. An incorrect but overconfident step may therefore be prioritized, while a correct but underconfident step may be explored later. Confidence contributes only an additional bounded penalty on top of the unit transition cost, while the independent depth–coverage heuristic and voting over multiple goal traces reduce dependence on any single confidence estimate.

2.4 Heuristic Design and Admissibility Guarantee

The heuristic estimates the remaining cost from a partial trace to a valid goal. In our implemented A* system, we use lightweight, interpretable search-control signals rather than a learned verifier. The heuristic combines depth and coverage: $h(n) = h_{\text{depth}}(n) + h_{\text{coverage}}(n)$ with

$$h_{\text{depth}}(n) = \omega_d \max(0, H_{\min} - d_n),$$
$$h_{\text{coverage}}(n) = \omega_c(1 - E(n)). \quad (4)$$

Here $H_{\min}$ is a task-level lower bound on required reasoning hops from the root, d_n is the depth of node n, $E(n) \in [0, 1]$ is the fraction of target entities or constraints already covered by the partial trace, and $\omega_d, \omega_c > 0$ are heuristic weights. Thus, h_{depth} encourages search to continue until a minimum reasoning depth is reached, while h_{coverage} favors traces that cover more answer-relevant entities or constraints. Let $c_{\min} = 1+(1-c_{\max})$ be the minimum possible transition cost under the path-cost definition. Since each transition has base cost 1 plus uncertainty penalty $1 - c_t$, every transition costs at least $c_{\min}$. The confidence and coverage terms are practical search-control signals, not calibrated uncertainty estimates or exact measures of evidential completeness.

Assumption 1: Minimum-hop lower bound. Any valid goal trace requires at least $H_{\min}$ reasoning steps from the root; equivalently, any non-goal node with $d_n < H_{\min}$ requires at least $H_{\min} - d_n$ further transitions.

Assumption 2: Non-goal continuation. Any non-goal node with $d_n \geq H_{\min}$ still requires at least one additional transition before reaching a goal.

Theorem 1: Conservative admissibility. Under Assumptions 1 and 2, if $\omega_d + \omega_c \leq c_{\min}$, then the depth-plus-coverage heuristic is admissible for all non-goal nodes: $h(n) \leq h^*(n)$ where $h^*(n)$ is the true minimum remaining path cost from node n to a valid goal.

Proof. The formal proof is provided in Section I.

Operational instantiation: a rule-based progress critic. We instantiate the heuristic as a lightweight, dataset-agnostic progress critic rather than an additional LLM verifier. For a partial state $n = (\tau_n, d_n)$, the critic assigns a progress score $c_{\text{rule}}(n) \in [0, 1]$, and A* uses the remaining-cost heuristic $h(n) = 1 - c_{\text{rule}}(n)$.

Concretely, the critic scores progress using the same two conservative signals as Eq. (4): minimum-hop completion and question-content coverage. Let E_q be the set of content units extracted from the question, including quoted strings, proper-name phrases, numbers, years, and non-stopword content terms, and let

$$E(n) = \frac{|\{e \in E_q : e \text{ appears in } \tau_n\}|}{|E_q|},$$

with $E(n) = 1$ if $E_q = \emptyset$. The critic score is

$$c_{\text{rule}}(n) = 1 - [\omega_d \max(0, H_{\min} - d_n) + \omega_c(1 - E(n))] \quad (5)$$

so that: $h(n) = \omega_d \max(0, H_{\min} - d_n) + \omega_c(1 - E(n))$.

Thus, depth and coverage are not alternatives to the critic-based heuristic; they are the internal scoring rules of a deterministic progress critic. This keeps A* lightweight and dataset-agnostic while prioritizing states that have made greater structural and content-level progress. Appendix A gives implementation details, admissibility analysis, and human validation.

Human validation of critic progress. We validate whether the rule-based critic behaves conservatively on intermediate reasoning states. We sample 500 partial states produced during A* search and ask human annotators to assign normalized progress scores. We then compare these scores with the critic-implied progress $c_{\text{rule}}(n) = 1 - h(n)$. The critic underestimates human-assessed progress on average (0.43 vs. 0.58, paired t-test $p < 0.001$), with an overestimation rate of 12.3% and maximum observed overestimation $\epsilon_{\max} = 0.14$. This supports the critic as a conservative search-control signal rather than an unconstrained verifier.

2.5 Routing Strategies

Given the complementarity between CoT and A*, we define routing functions $\mathcal{R} : \mathcal{Q} \to \{\text{CoT}, A^*\}$. We distinguish two families.

Output-based routers. They require generating traces before deciding. *Semantic entropy* (Farquhar et al., 2024): routes high-entropy (under self-consistency) instances to A*. *LLM-as-critic*: on disagreements, a critic model (DeepSeek-R1-Distill-Qwen-14B, chosen for architectural distinctness from the LLaMA-3.1-8B backbone and local executability) compares both traces:

$$\text{verdict} = \mathcal{M}_{\text{critic}}(\text{Trace}_{\text{CoT}}, \text{Trace}_{A^*}, q, C, \mathcal{O}) \quad (6)$$

The LLM-as-Critic baseline is a post-hoc pairwise trace selector and is distinct from the rule-based progress critic used internally during A* search. The latter is deterministic, observes a partial reasoning state, and is used only for search-time prioritization.

Pre-generation routers. It decides before any generation. *Random forest*: trained on hand-crafted features (question length, connectives, negation). *Topology router*: XGBoost on the 13 topology features, requiring no LLM inference at routing time. It is a diagnostic test of whether the CoT/A* boundary is predictable from problem structure.

The router is evaluated using stratified 5-fold cross-validation with seed 42, performed separately within each benchmark. In each fold, the router is trained on four folds and evaluated only on the held-out fold. The reported benchmark accuracy combines predictions from the five held-out folds.

At inference time, the router receives only the 13 features computed from the input before generation. It does not observe the CoT or A* trace, model confidence, generated answers, correctness labels, gold answers, or an explicit dataset identifier. This protocol evaluates within-dataset generalization to unseen instances from the same benchmark distribution; it does not by itself establish transfer to an unseen benchmark.

The present binary router is deliberately restricted to the paper's central comparison: a single linear CoT trajectory versus explicit heuristic-guided search over partial reasoning states. Self-consistency instead samples multiple independent complete CoT trajectories and aggregates their answers, without maintaining a frontier, backtracking, or prioritizing partial states with a heuristic. Therefore, the current router should not be interpreted as identifying the globally best inference strategy among CoT, self-consistency, A*, and Tree-of-Thought.

2.6 Reasoning Topology Features based Routing

We use *reasoning topology* operationally rather than in the mathematical-topology or topological-data-analysis sense. It refers to the branching, depth, and path-comparison structure induced by a reasoning problem: whether the problem can be solved through a largely linear, globally coherent trajectory or instead requires preserving, comparing, or eliminating multiple partial reasoning paths before committing to an answer.

To characterize the structural demands of a reasoning problem *before* any model generation, we define 13 pre-generation features computed directly from the input text. These include surface length features: question length, context length, and number of context sentences; syntactic and discourse cues: logical connectives, negation markers, comparatives, and question marks; reasoning-structure features: estimated hop count from dependency links, contextual density measured as entities per sentence, distractor count, and estimated reasoning depth; and task-format features such as the number of answer options. We also define a derived *branching coefficient*,

$$\mathrm{BC} = \frac{c_{\mathrm{sent}} \times \mathrm{hop}}{q_{\mathrm{len}}} \tag{7}$$

where c_{sent} is the number of context sentences, hop is the estimated reasoning-hop count, and q_{len} is the question length. BC should be interpreted as an operational proxy for multi-hop branching structure rather than as a formal topological quantity. It summarizes how much multi-hop branching structure the input affords relative to question length. where $\mathrm{c_sent}$ is the number of context sentences, hop is the estimated reasoning-hop count, and $\mathrm{q_len}$ is the question length used to normalize branching complexity. These features require no model inference and are used for both topology clustering (§3.3) and pre-generation routing (§3.4).

Some of these features, including input length, connective counts, and answer-format information, may also capture surface-level or benchmark-specific properties. We therefore interpret the resulting cluster and routing analyses as supporting evidence for a structural explanation rather than as proof that all dataset-specific signal has been eliminated.

3 Results

We evaluate on five benchmarks using three models: i) LLaMA-3.1-8B and ii) Qwen-0.5B, iii) Qwen-2.5-14B and we use DeepSeek-R1-Distill-Qwen-14B as the LLM critic. The following benchmarks are used: StrategyQA (Geva et al., 2021) tests multi-step commonsense reasoning with boolean answers. LogicQA (Liu et al., 2020) tests formal logical reasoning with four-way multiple choice. HotpotQA (Yang et al., 2018) tests multi-hop question answering with free-form answers. Also we use GSM8k (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) as standard mathematical benchmarks. We report the exact accuracies from the current experimental pipeline, and the oracle corresponds to instance-wise selection of the correct answer between CoT and A*.

3.1 Quantitative Analysis

Main results. Table 1 shows that no single reasoning strategy dominates. A* is substantially stronger than CoT on StrategyQA (+10.28 pp) and HotpotQA (+3.30 pp), while CoT has a slight edge on LogicQA (+0.93 pp). Yet the oracle sits far above either strategy—improving by +9.79, +13.51, and +14.10 points respectively—confirming strong complementarity. The presence of substantial CoT-only and A*-only cells in Table 2 shows the methods implement genuinely different computations, motivating the topology analysis (Section 3.3) and routing (Section 3.4).

This complementarity is not limited to one model or dataset family. On Qwen-0.5B, A* gives a large gain on HotpotQA (30.00 vs. 8.40), but not on StrategyQA or LogicQA, suggesting that search helps small models mainly when the task structure rewards exploration. On Qwen-2.5-14B, A* is especially effective on HotpotQA (83.33 vs. 60.19) and gives smaller gains on GSM8K and MATH, while self-consistency is strongest on Strat-

Dataset	LLaMA-3.1-8B				Qwen-0.5B				Qwen-2.5-14B			
	CoT	A*	SC	ToT	CoT	A*	SC	ToT	CoT	A*	SC	ToT
StrategyQA	64.52	**74.80**	71.22	72.24	53.60	53.40	**54.21**	52.24	76.02	75.56	**80.06**	78.17
LogicQA	45.78	44.85	46.10	**46.21**	**24.00**	20.20	22.10	21.19	57.91	58.12	**59.19**	55.12
HotpotQA	76.80	**80.10**	78.21	78.45	8.40	**30.00**	12.21	22.17	60.19	**83.33**	65.25	72.14
GSM8K	85.04	86.12	**86.90**	60.00	49.60	**50.70**	40.90	38.60	93.01	**94.88**	94.08	93.78
MATH	78.00	79.10	**79.50**	40.10	34.40	35.20	**46.60**	33.10	79.11	**80.78**	80.04	79.50

Table 1: Consolidated results across reasoning and math benchmarks. Scores denote accuracy (%). SC denotes Self-Consistency and ToT denotes Tree-of-Thought. Within each dataset–model block, the best single-method score among CoT, A*, SC, and ToT is bolded, and the second-highest score is underlined.

egyQA and LogicQA. Overall, search is most useful for multi-hop or branch-heavy inputs, whereas sampling-based or linear methods remain competitive on more compact reasoning problems.

Router performance. We evaluate all routing strategies on LLaMA-3.1-8B to test whether a router can choose between CoT and A* more effectively than either fixed strategy. Table 4 shows that routing closes only part of the oracle gap, but the topology-aware router is consistently strongest: on the QA benchmarks, it improves over the stronger fixed baseline by +2.55 pp on StrategyQA, +3.23 pp on LogicQA, and +2.31 pp on HotpotQA, while recovering 26.1% of the oracle gap on StrategyQA. The pattern also extends to math, where the topology router reaches 91.10% on GSM8K and 81.52% on MATH, outperforming CoT, A*, semantic entropy, LLM-as-critic, LLM-as-judge, and random-forest routing. This contrast is important: trace-based routers yield only modest gains and sometimes underperform A* alone, whereas the topology router uses only 13 pre-generation structural features. Thus, the CoT–search decision is better predicted from problem topology than from retrospective evaluation of completed reasoning traces.

Dataset	Both correct	CoT only	A* only	Both wrong
StrategyQA	54.73	9.79	20.07	15.41
LogicQA	31.34	14.44	13.51	40.71
HotpotQA	62.70	14.10	17.40	5.80

Table 2: Instance-level complementarity decomposition (%). "CoT only" means CoT is correct and A* is wrong; "A* only" means the reverse.

Instance-level decomposition. Table 2 makes the complementarity concrete. HotpotQA is especially striking: only 5.8% of instances are missed by both strategies, while 31.5% lie in a regime where exactly one strategy is correct. StrategyQA also exhibits sizable disagreement. LogicQA is harder overall, as shown by the 40.7% "both wrong", but even there each strategy recovers failures of the other.

3.2 Qualitative Analysis

We manually inspect representative CoT–A* disagreement cases from StrategyQA, LogicQA, and HotpotQA and group them into recurring *reasoning regimes*. The qualitative evidence supports the quantitative complementarity in Table 2: A* helps when the input induces several plausible partial trajectories, whereas CoT remains preferable when the solution is naturally linear and executable in one coherent pass. Appendix B provides the corresponding examples, organized into A* success cases, CoT success cases, A* failure modes, and human trace-quality analysis.

When A* helps. A* is most useful when the reasoning topology is *branching*: the model must compare, preserve, or eliminate multiple partial hypotheses before committing. Appendix B shows three such regimes. In *non-linear evidence aggregation* (Appendix B.1), decisive facts do not form a single fluent chain; CoT may follow a locally plausible narrative, while search keeps alternatives open until a globally better path appears. In *hypothesis elimination*, especially multiple-choice LogicQA, the task is less to derive one proof than to rule out plausible distractors; search can distribute budget across options, whereas CoT may commit to the first surface-compatible choice. In *multi-hop composition*, especially HotpotQA, search preserves competing entity or relation chains until the bridge entity is found, improving evidence traversal rather than adding new knowledge.

Why these gains arise. The common mechanism is *deferred commitment*. CoT commits to a single

left-to-right trajectory, which is efficient when the early steps are correct but brittle when they are only partially informative or misleading. A* treats intermediate thoughts as expandable states, allowing the model to back away from an initially promising but suboptimal path. Its advantage is therefore not generic extra computation, but better allocation of reasoning effort across competing partial solutions, as illustrated by the A* success cases in Appendix B.1.

When CoT remains preferable. Search hurts when the problem does not require exploration. Appendix B.2 shows two such regimes. In *direct factual reasoning*, the answer follows from a small set of stable facts or a short deterministic comparison; a single fluent chain is sufficient, while A* may introduce irrelevant distinctions. In *globally coherent argumentation*, especially some LogicQA cases, the solution requires maintaining one abstract frame across the whole argument. CoT can preserve this global frame in a continuous explanation, whereas A* may fragment it into locally plausible but globally weaker steps.

Characteristic failure modes of A*. The A* errors are systematic rather than random, and Appendix B.3 gives representative cases. The dominant failure is *overthinking simple problems*: when direct recall or a short inference suffices, search may add unnecessary distinctions and drift from the obvious answer. A second failure is *spurious associative expansion*, where a weak semantic link is elaborated despite limited causal relevance. A third is *budget waste through repetition or near-duplicate expansions*, where search revisits the same intermediate fact instead of exploring genuinely new possibilities. These cases show that search changes the *shape* of reasoning, not only its computational cost; when branching is unnecessary, this changed shape can reduce correctness.

Takeaway. The CoT–A* tradeoff is topological: A* helps when the problem structure warrants exploration, while CoT is stronger when direct, coherent commitment is enough.

3.3 Explaining Complementarity Through Topology

We formalize the qualitative intuition using the topology features from Section 2.6. Applying k-means (k=4) to the 13 features across all instances yields four clusters (Table 3). A* advantage increases from -0.4 pp (C1) to +11.4 pp (C4), with $R^2 = 0.91$. Each cluster contains instances from multiple benchmarks, while A* advantage increases with branching complexity. This provides evidence that structural variation helps explain the CoT/A* boundary beyond dataset identity alone, although the hand-crafted proxies may still retain some benchmark-specific signal.

Cluster	Profile	BC	A*	CoT	Interpretation
C1	Linear, short	1.28	81.7	82.1	CoT sufficient
C2	Mid-branch	4.24	73.5	66.2	selection critical
C3	Complex, noisy	4.02	52.3	49.1	both often fail
C4	Deep-branch	6.49	69.8	58.4	A* largest gain

Table 3: Reasoning topology clusters. BC = branching coefficient. A* advantage grows monotonically with branching complexity.

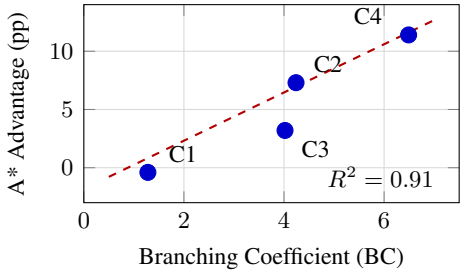


Figure 2: A* advantage (pp) vs. branching coefficient across topology clusters. Linear fit: $R^2 = 0.91$.

Prescriptive decision rule. The cluster analysis yields a practical decision rule: *C1 (linear)*: use CoT; search overhead is unnecessary. *C2 (mid-branch)*: route adaptively; this is where router quality matters most. *C3 (complex, noisy)*: neither method is reliable; retrieval augmentation or stronger models may be needed. *C4 (deep-branch)*: use A*; search has the clearest advantage.

Topic modelling validates topology. As independent validation, we apply Non-negative Matrix Factorisation (NMF, k=8) on TF-IDF representations of all paired evaluation questions. A* dominates in 6 of 8 topic categories covering 66.2% of the corpus—these share multi-hop entity chains (entertainment, historical events, geography, biography, politics). CoT-dominant topics (33.8%) correspond to single-step commonsense categories (everyday reasoning, hypothetical scenarios). Full per-topic results with examples in Appendix F.

3.4 Topology-Aware Routing

Table 4 compares all routers.

Why pre-generation routing outperforms

Router	Input	Str.QA	Log.QA	Hot.QA	GSM8K	MATH
A* only	–	74.80	44.85	80.10	86.12	79.10
CoT only	–	64.52	45.78	76.80	85.04	78.00
Sem. Entropy	traces	74.37	47.47	81.30	86.67	78.24
LLM-as-Critic	traces	74.19	47.16	80.21	87.28	79.13
LLM-as-Judge	traces	73.28	47.21	80.87	85.65	78.23
Random Forest	features	76.25	48.20	81.25	87.43	80.12
Topology Router	**13 feat.**	**77.35**	**49.01**	**82.41**	**91.10**	**81.52**
Oracle	–	84.59	59.29	94.20	96.14	84.17

Table 4: Router comparison on LLaMA-3.1-8B. Scores denote accuracy (%). The topology router uses only pre-generation structural features and outperforms trace-based routers. All QA router improvements over CoT are statistically significant under McNemar's test ($p < 0.005$). LLM-as-Judge and LLM-as-Critic differ only in prompt design; prompts are provided in Appendix Figure 3.

Metric	A*	CoT
Average fluency score	0.725	0.673
Cases where more fluent	62.0%	38.0%
More-fluent but wrong	38.0% of disagreements	

Table 5: Fluency-Correctness Asymmetry on disagreement cases. Trace fluency does not predict correctness.

post-hoc routing. The topology router observes no generated traces, yet outperforms routers that inspect both full reasoning traces. This reverses the intuitive expectation that seeing full traces should be more informative. The bottleneck is often not judging which completed trace looks better, but recognizing before generation which computation the problem requires.

Fluency–Correctness Asymmetry. In the sampled disagreement cases, we observe an empirical limitation of output-based routing: in 38% of sampled disagreement cases, the more fluent trace is incorrect (Table 5). A trace selector that rewards coherence can therefore prefer the wrong computation. This pattern provides one possible explanation for why the topology router outperforms the evaluated LLM-as-critic.

3.5 Sensitivity, Compute, and Failure Analysis

Sensitivity analysis. On a stratified 300-question HotpotQA sample (seed=42), we sweep the node budget $B \in \{5, 10, 15, 20, 30\}$ and branching factor $K \in \{1, 2, 3, 4\}$ (full tables in Appendix D). Accuracy improves from B=5 (73.0%) to B=20 (78.7%), then drops at B=30 (76.0%), indicating diminishing returns. The default B=15 (78.0%) lies within 0.7 pp of the peak. Even K=1 (no real branching) achieves 77.7%, confirming graceful degradation and remaining competitive self-consistency.

Compute–accuracy tradeoff. A* at the default (K=3, B=15) uses ∼12,452 measured tokens/question (∼24× CoT's ∼512). A budget-reduced variant (K=1) achieves competitive accuracy at 3,372 tokens/question (∼7× CoT). A* is substantially more expensive; its use is justified only when topology predicts branching is useful, motivating pre-routing as a compute-saving mechanism (full table in Appendix E).

Failure analysis. Among A* failures on instances CoT solves correctly, 91.1% are *overthinking* (search continues past correct evidence, accumulating distractors) and 8.9% are *premature commitment* (search locks onto an early branch). CoT fails by committing too early; A* fails by continuing too long. The same branching mechanism that helps on multi-hop problems hurts on linear ones (Examples in Appendix B.3).

Human evaluation. We conduct a human evaluation of reasoning-trace quality using the criteria in §B.4. Two post-graduate annotators with experience in LLM reasoning research each labeled 100 instances from StrategyQA and LogicQA. Across datasets and metrics, Cohen's κ ranges from 0.59 to 0.68, indicating fair agreement. Two annotators labeled 100 disagreement instances per dataset on *coverage* (1–3) and *relevance* (1–3), with Cohen's κ: 0.59–0.68. The correct method consistently achieves higher coverage (e.g., StrategyQA: CoT-correct 3.00 vs. A* 1.20; A*-correct 2.71 vs. CoT 1.63), confirming that disagreements are driven by coverage of the required reasoning path rather than surface quality (full results in Appendix B.4).

4 Related Work

CoT and structured reasoning. Chain-of-thought prompting (Wei et al., 2022) reasons through a single linear trace, while self-consistency (Wang et al., 2023), Tree-of-Thoughts (Yao et al., 2023), and Graph-of-Thoughts (Besta et al., 2024) add sampling, tree search, or graph-structured deliberation. Prior work mostly asks whether richer reasoning structures improve aggregate accuracy, and recent work shows that CoT itself is not uniformly beneficial. We ask the complementary question: when does moving from linear reasoning to search help, and when does it hurt?

Search-augmented reasoning. MCTS-style reasoning (Hao et al., 2023), beam search (Xie et al., 2023), LATS (Zhou et al., 2024), MCTSr (Zhang et al., 2024), and rStar-Math (Guan et al., 2025) demonstrate the value of explicit exploration. However, aggregate gains alone do not reveal which inputs benefit from search. We use A* as a controlled diagnostic and show that search gains are topology-dependent: branching, multi-hop, and hypothesis-elimination problems benefit most, while simpler linear cases can degrade.

Over-exploration and search failures. Work on overthinking and excessive reasoning often treats over-exploration as an efficiency issue (Sui et al., 2025; Ma et al., 2025; Wang et al., 2025). We show it is also a correctness issue: 91.1% of A* failures on CoT-correct instances arise from overthinking, where search continues beyond sufficient evidence, or revisits near-duplicate states.

Routing and critic limitations. Semantic entropy (Farquhar et al., 2024), LLM-as-judge methods (Gu et al., 2026), and process supervision (Lightman et al., 2024) rely on generated traces to assess reasoning quality, while mixture-of-experts models (Shazeer et al., 2017) route between neural modules. We instead route between *reasoning algorithms* before generation, using input topology. This is crucial because our Fluency–Correctness Asymmetry shows that fluent traces can be wrong: In 38% of sampled disagreement cases, the more fluent trace is incorrect. Our contribution is a diagnostic framework for deciding *when* search should be used.

Method	Structure	Front.	Heur.	Pre-rte	Limitation
CoT	linear	✗	✗	✗	premature commit
SC	multi-linear	✗	✗	✗	unguided sampling
ToT	tree/BFS	✓	eval.	✗	evaluator entangled
MCTS	tree+rollout	✓	value	✗	many design choices
A* (ours)	priority	✓	✓	✓	higher compute

Table 6: Methodological comparison of reasoning strategies. Front. = explicit frontier; Heur. = heuristic guidance; Pre-rte = pre-generation routing studied. A* isolates the role of heuristic-guided search without entangling evaluator design or rollout policy.

5 Conclusion

Across five benchmarks and three model scales, we find that CoT and A* are complementary rather than hierarchical: the gap between the oracle and the best single method is consistently large. The key explanatory variable is branching coefficient, not dataset identity—a structural finding that holds across both QA and math domains. Perhaps most surprisingly, a router that sees only the raw input (no traces) outperforms critics that inspect both full reasoning chains, because trace fluency misleads post-hoc judges in a systematic way. These results suggest that the next step for adaptive inference is not better trace evaluation but better problem-structure recognition. Concrete next steps include learned topology embeddings.

Limitations

Our findings are conditioned on a single budgeted A* implementation; a different search policy (MCTS, beam search) could shift where the complementarity boundary lies. The 13 topology features are hand-designed proxies—interpretable but unlikely to capture the full latent structure of reasoning difficulty. They also rely on English-language parsing tools, limiting direct transfer to other languages. On the cost side, A* at default settings uses $\sim 24\times$ the tokens of CoT, and our token accounting covers only a 300-question ablation sample. Domain coverage now includes math (GSM8K, MATH) but long-form generation remain untested. Finally, the topology router is an XGBoost classifier on fixed features; a learned router that jointly optimizes routing and search would be a natural next step. Some of the topology features, particularly raw input length, connective counts, and answer-format features, may retain benchmark-specific surface information. The

within-dataset router evaluation therefore demonstrates generalization to unseen instances from the same benchmark distribution, but does not establish transfer to an unseen benchmark. The current CoT–A* router is accuracy-oriented and deliberately restricted to the linear-versus-search comparison; a broader cost-aware router over CoT, self-consistency, and search remains an important extension.

Acknowledgements

Dibyanayan Bandyopadhyay gratefully acknowledges support from the Prime Minister's Research Fellowship (PMRF) under Grant No. 2703564. The authors also thank the anonymous reviewers for their constructive feedback and valuable suggestions.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

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

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2026. A survey on llm-as-a-judge. *The Innovation*, 7(6):101253.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rStar-math: Small LLMs can master math reasoning with self-evolved deep thinking. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 20640–20661. PMLR.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035, Vienna, Austria. Association for Computational Linguistics.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*.

Ante Wang, Linfeng Song, Ye Tian, Dian Yu, Haitao Mi, Xiangyu Duan, Zhaopeng Tu, Jinsong Su, and Dong Yu. 2025. Don't get lost in the trees: Streamlining LLM reasoning by overcoming tree search exploration pitfalls. In *Proceedings of the 63rd Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 23946–23959, Vienna, Austria. Association for Computational Linguistics.

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

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

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2023. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems*, volume 36, pages 41618–41650. Curran Associates, Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *Preprint*, arXiv:2406.07394.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. 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.

A Rule-Based Progress Critic and Heuristic Analysis

Implementation. Section 2.4 writes the A* heuristic as $h(n) = 1 - c_{\text{rule}}(n)$, where c_{rule} is a deterministic progress critic. The critic does not call an external LLM and does not judge the factual truth of a partial trace. It computes progress from two generic signals: whether the trace has reached a minimum reasoning depth and whether it covers question-level content units. The same extraction rule is used across datasets: quoted spans, proper-name phrases, numbers, years, and non-stopword content terms are extracted from the question and matched against the partial trace.

Algorithm 1 Dataset-Agnostic Rule-Based Progress Critic

Input: Question q; partial reasoning state $n = (\tau_n, d_n)$; minimum-hop lower bound $H_{\min}$; weights ω_d, ω_c

Output: Heuristic value $h(n)$

Step 1: Extract question content units.
Initialize $\mathcal{E}_q \leftarrow \emptyset$.
Add all quoted strings in q to $\mathcal{E}_q$.
Add all proper-name phrases in q to $\mathcal{E}_q$.
Add all numbers and years in q to $\mathcal{E}_q$.
Add all non-stopword content terms of length greater than three to $\mathcal{E}_q$.

Step 2: Compute entity/content coverage.
if $\mathcal{E}_q = \emptyset$ **then**
    $E(n) \leftarrow 1$
else
    $C(n) \leftarrow \{e \in \mathcal{E}_q : e \text{ appears in } \tau_n\}$
    $E(n) \leftarrow \frac{|C(n)|}{|\mathcal{E}_q|}$
end if

Step 3: Compute remaining structural progress.
$r_{\text{depth}}(n) \leftarrow \max(0, H_{\min} - d_n)$
$h_{\text{depth}}(n) \leftarrow \omega_d r_{\text{depth}}(n)$

Step 4: Compute remaining content progress.
$h_{\text{coverage}}(n) \leftarrow \omega_c(1 - E(n))$

Step 5: Return total heuristic.
$h(n) \leftarrow h_{\text{depth}}(n) + h_{\text{coverage}}(n)$
return $h(n)$

Admissibility and consistency. The rule-based progress critic induces the remaining-cost heuristic $h(n) = 1 - c_{\text{rule}}(n)$, which is exactly the depth–coverage quantity in Eq. (4). Under the minimum-hop and non-goal-continuation assumptions, this heuristic is admissible when its weights are bounded by the minimum transition cost. In

our implementation, $H_{\min} = 2$, $\omega_d = \omega_c = 0.3$, and $c_{\max} = 0.99$, so

$$c_{\min} = 1 + (1 - c_{\max}) = 1.01.$$

The largest possible heuristic value at the root is

$$2\omega_d + \omega_c = 0.9,$$

which is below the minimum two-step path cost $2c_{\min} = 2.02$. After one step, the heuristic is at most $\omega_d + \omega_c = 0.6 < c_{\min}$, and after the minimum-hop threshold the coverage term is at most $\omega_c = 0.3 < c_{\min}$. Hence $h(n)$ lower-bounds the remaining path cost under the stated assumptions and is admissible. It is also consistent, since a single transition can reduce the heuristic by at most $\omega_d + \omega_c = 0.6$, which is smaller than the minimum transition cost $c_{\min} = 1.01$.

B Qualitative Examples

This appendix provides the detailed examples corresponding to the qualitative conclusions summarized in Section 3.2.

B.1 Patterns of A* Success and CoT Failure

We identify three recurring settings where A* succeeds while CoT fails.

Group 1: Non-linear evidence aggregation. These are questions where the answer requires combining several facts that do not naturally form a single left-to-right narrative.

StrategyQA — Non-linear Evidence Aggregation

Q: *Is shrimp scampi definitely free of plastic?* (Gold: NO)

CoT (× predicts YES): Proceeds through five sequential steps analyzing harvesting, processing, cooking, ingredients, and regulations. Each step independently concludes no significant plastic risk. The linear structure prevents cross-referencing marine biology evidence with the food preparation chain.

A* (✓ predicts NO): Step 1 directly identifies microplastic bioaccumulation in shrimp tissue. Step 2 notes that cooking does not remove microplastics. The heuristic guides the search toward the most relevant evidence node, avoiding the exhaustive but misleading step-by-step analysis.

LogicQA — Non-linear Evidence Aggregation

Q: *Based on the above statement [spatial arrangement of communities around Civic Park], which of the following can be derived?* (Gold: A)

CoT (× predicts D): Analyzes spatial relationships sequentially, accumulating errors in relative positioning. By the fourth step, the accumulated imprecision leads to selecting an incorrect spatial relationship.

A* (✓ predicts A): Explores multiple spatial constraint combinations in parallel. The search finds a consistent assignment by simultaneously satisfying the "southwest of" and "southeast of" constraints, correctly deriving that Civic Park is north of the administrative service area.

Group 2: Hypothesis elimination. A* is also effective when the problem naturally decomposes into evaluating or eliminating alternatives.

LogicQA — Hypothesis Elimination

Q: *Which of the following is most similar to the argument: "All landscape rooms can see the landscape, but Li Wenbing's family can't see the landscape. Therefore, Li Wenbing's family is not a landscape room."* (Gold: D)

CoT (× predicts C): Identifies the logical structure (modus tollens) but then incorrectly maps it to option C (a modus ponens structure) due to surface similarity.

A* (✓ predicts D): Separately evaluates options through different branches. One branch identifies the negative inference pattern in the premise and correctly maps it to option D ("Anyone who meets conditions can apply; Sun Wen did not apply; therefore Sun Wen did not meet conditions"), which preserves the modus tollens structure.

Group 3: Multi-hop question answering. HotpotQA highlights the value of search when the answer depends on chaining facts across sources.

HotpotQA — Multi-hop Chaining

Q: *What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?* (Gold: Chief of Protocol)

CoT (× predicts "Ambassador"): Correctly identifies Shirley Temple as the actress but then retrieves an incorrect government position from parametric memory, confusing ambassadorial roles with her protocol role.

A*: Explores multiple evidence paths for Shirley Temple's government career. While the trace is imperfect, the tree search structure allows it to consider multiple candidate positions before committing.

B.2 Patterns of CoT Success and A* Failure

We identify two common settings where CoT remains better than search.

Group 1: Direct factual reasoning. When the answer follows directly from familiar facts, search can introduce unnecessary branching and semantic drift.

StrategyQA — Direct Factual Reasoning

Q: *Is a Boeing 737 cost covered by Wonder Woman (2017 film) box office receipts?* (Gold: YES)

CoT (✓ predicts YES): Directly retrieves the production budget (∼$150M), box office (∼$822M), and Boeing 737 cost (∼$100M), concluding that the receipts exceed the aircraft cost.

A* (× predicts NO): The first search step focuses on whether box office receipts are "associated with" aircraft purchases, misreading the question as one about financial transactions rather than magnitude comparison.

StrategyQA — Direct Factual Reasoning

Q: *Is average number of peas in a pod enough commas for a billion?* (Gold: YES)

CoT (✓ predicts YES): Correctly identifies that a billion (1,000,000,000) requires three commas and that a pea pod usually contains 7–9 peas, so the answer is yes.

A* (× predicts NO): Incorrectly states that a billion has 12 zeros and becomes confused by the scale comparison.

Group 2: Fluent world knowledge and argumentation. When the problem is best solved by a smooth, coherent narrative, CoT often remains more reliable than fragmented search steps.

LogicQA — Fluent Argumentation

Q: *Which of the following is the best argument against the claim that rich Chinese are transferring property abroad?* (Gold: B)

CoT (✓ predicts B): Systematically compares the answer options and identifies the strongest counterargument.

A* (× predicts A): The first search step identifies option A as relevant, and the heuristic keeps the search near that branch without sufficiently comparing the full option set.

B.3 A* Failure Modes on Simple Cases

We observe three recurrent failure modes for A* on examples that CoT solves easily.

Failure mode 1: Overthinking simple questions.

StrategyQA — Overthinking

Q: *Does Disney have an ice princess?* (Gold: YES)

A* (× predicts NO): Step 1 correctly identifies Elsa from *Frozen* as having ice powers. Step 2 notes that she is called "The Ice Queen" in promotional material. The search then turns the easy recall problem into an unnecessary semantic distinction between "queen" and "princess," yielding the wrong answer.

Failure mode 2: Spurious correlations in search.

StrategyQA — Spurious Correlation

Q: *Is Menthol associated with Thanksgiving?* (Gold: NO)

A* (× predicts YES): Step 1 associates menthol with cough drops used in cold weather. Step 2 associates Thanksgiving with cold weather in North America. Step 3 turns these weak associations into a spurious chain, effectively reasoning menthol → cold weather → Thanksgiving.

Failure mode 3: Step repetition.

StrategyQA — Step Repetition

Q: *Will the Albany in Georgia reach a hundred thousand occupants before the one in New York?* (Gold: NO)

A* (× predicts YES): Step 2 states the population of Albany, New York (∼98,000). Step 3 largely repeats the same fact. The repetition wastes node budget that should have been used to compare growth trajectories.

B.4 Human evaluation

We conduct a human evaluation of reasoning-trace quality using the criteria in §B.4. Two post-graduate annotators with experience in LLM reasoning research each labeled 100 sampled comparison instances per dataset from StrategyQA and LogicQA.

Across datasets and metrics, Cohen's κ ranges from 0.59 to 0.68, indicating fair agreement. The evaluation samples were different.

Annotation criteria. *Coverage* measures whether a trace reaches the required inferential path and supports the final conclusion. We use a 3-point scale: 1 if the reasoning is mostly incorrect and does not reach the conclusion, 2 if it is mostly correct but still fails to reach the conclusion, and 3 if both the reasoning and conclusion are correct.

Relevance measures how focused the trace is on concepts actually needed to answer the question. We score it coarsely on a 3-point scale based on the proportion of relevant steps, where relevant steps remain on-topic and avoid introducing unnecessary concepts.

Algorithm 2 Budgeted A* Search over Reasoning Traces

Input: Problem $\mathcal{Q} = (q, C, \mathcal{O})$, model $\mathcal{M}_\theta$, depth limit $D_{\max}$, node budget B, candidates N_c

Output: Final answer from weighted vote over goal traces

Initialize root $n_0 \leftarrow (\emptyset, 0)$
$g(n_0) \leftarrow 0$
$h(n_0) \leftarrow h(n_0)$ from Eq. (4)
OPEN $\leftarrow \{n_0\}$ ▷ priority queue on $f(n) = g(n) + h(n)$
VISITED $\leftarrow \emptyset$, GOALS $\leftarrow \emptyset$, explored $\leftarrow 0$
while OPEN $\neq \emptyset$ and explored $< B$ **do**
  $n \leftarrow$ OPEN.pop_min()
  if sig$(n) \in$ VISITED **then**
    continue
  end if
  VISITED $\leftarrow$ VISITED $\cup \{\text{sig}(n)\}$
  explored $\leftarrow$ explored $+1$
  if ISGOAL(n) **then**
    GOALS $\leftarrow$ GOALS $\cup \{n\}$
    continue
  end if
  if $d_n \geq D_{\max}$ **then**
    continue
  end if
  $\{(s^{(j)}, c^{(j)})\}_{j=1}^{N_c} \leftarrow$ GENERATESTEPS$(\mathcal{M}_\theta, q, C, \mathcal{O}, \tau_n, N_c)$
  for each $(s^{(j)}, c^{(j)})$ **do**
    $n' \leftarrow (\tau_n \oplus s^{(j)}, d_n + 1)$
    $g(n') \leftarrow g(n) + 1 + (1 - c^{(j)})$
    OPEN.push$(n', g(n') + h(n'))$
  end for
end while
return weighted vote over GOALS

Relevance is less diagnostic. In StrategyQA, CoT remains more relevant than A* when both are wrong (2.82 vs. 1.55), while in LogicQA both methods stay moderately relevant even when incorrect (2.76 for CoT, 2.53 for A*). Thus, staying on-topic is not enough: a trace may remain relevant yet still miss the decisive inferential steps. Overall, the annotations indicate that the main difference between CoT and A* on disagreement cases lies in *which method better covers the decisive reasoning steps*, while relevance is only weakly associated with correctness.

Interpretation. Refer to Table 8. Across both datasets, the method that is correct on a disagreement case consistently receives a much higher *cov-*

LLM-as-Judge

Purpose. Compare two candidate reasoning trajectories and decide which one solves the query more *faithfully* and *correctly*.

Inputs.

- Context
- Query
- Path A reasoning trace
- Path B reasoning trace

Evaluation criteria.

- **Faithfulness:** uses only the provided context and avoids invented constraints
- **Logical validity:** each step follows causally and coherently from prior steps
- **Grounded conclusion:** the final answer directly resolves the query from the established reasoning

Required output. The model first reasons internally about both traces, then returns only the winning path in constrained XML:

`<winner>A</winner>` or `<winner>B</winner>`

LLM-as-Critic

Purpose. Diagnose the weaknesses of two reasoning traces and determine which trace is better overall.

Inputs.

- Question
- Answer options
- Trace A with predicted answer
- Trace B with predicted answer

Evaluation criteria.

- whether the reasoning actually entails the predicted answer
- whether there are logical fallacies or unsupported jumps
- whether any claims are hallucinated or ungrounded

Required output. The model returns a structured JSON critique with flaws for both traces and the preferred trace:

```
{"trace_A_flaws": "...",
 "trace_B_flaws": "...",
 "better_trace": "A/B"}
```

Figure 3: LLM-based routing prompts. The *judge* prompt is verdict-oriented: it compares two reasoning paths under faithfulness and logical soundness criteria and outputs only the winner. The *critic* prompt is diagnosis-oriented: it identifies flaws in each trace before selecting the better one in structured JSON form.

Dataset	# disagreement cases	Annotation sample
StrategyQA	587	100
LogicQA	332	100

Table 7: Size of the disagreement pool used for human-style analysis. We consider only instances where CoT and A* produce different final predictions. In single-label QA, the *both-right* bucket is therefore expected to be empty, which is confirmed in both datasets.

erage score than the method that is wrong. On StrategyQA, when CoT is correct and A* is wrong, CoT achieves average coverage 3.00 versus 1.20 for A*; when A* is correct and CoT is wrong, A* scores 2.71 versus 1.63 for CoT. The same pattern holds on LogicQA (3.00 vs. 1.86 for CoT-correct cases, and 2.77 vs. 1.88 for A*-correct cases). This suggests that disagreement cases are driven mainly by *coverage of the required reasoning path*, not by superficial stylistic differences.

C FAQs

Q1. Why are gains on HotpotQA larger than StrategyQA? HotpotQA explicitly rewards multi-hop evidence aggregation; StrategyQA is often solvable linearly.

Q2. Why include LogicQA where A* underperforms? LogicQA stresses formal sequential reasoning where branching can interfere, giving a more honest picture.

Q3. How reliable are self-reported confidence scores? They are pragmatic ranking signals (not calibrated probabilities) biasing search toward shorter, more confident traces.

Q4. How should readers interpret the admissibility result? The formal proof (Appendix I)) applies to the depth–coverage heuristic used for StrategyQA and HotpotQA under the stated assumptions. The LogicQA-specific heuristic is evaluated empirically but is not covered by the theorem.

Q5. Would ensemble voting capture the same complementarity? Only partially. Self-consistency recovers diversity but does not guide exploration toward promising reasoning states.

Q6. Is goal detection based on answer patterns brittle? Somewhat. Pattern matching is fast and reproducible but can miss valid conclusions phrased differently.

Dataset	Correctness group	n	CoT Cov.	CoT Rel.	A* Cov.	A* Rel.
StrategyQA	both wrong	11	1.00	2.82	1.09	1.55
	CoT right, A* wrong	51	3.00	2.57	1.20	2.16
	CoT wrong, A* right	38	1.63	2.47	2.71	2.00
LogicQA	both wrong	38	1.79	2.76	1.89	2.53
	CoT right, A* wrong	36	3.00	2.81	1.86	2.47
	CoT wrong, A* right	26	1.88	2.81	2.77	2.42

Table 8: Human-style annotation on 100 sampled comparison cases per dataset cases per dataset. "Cov." denotes reasoning coverage (1–3), and "Rel." denotes relevance (1–3). Coverage measures whether the reasoning substantially reaches a correct conclusion; relevance measures the proportion of steps that stay on-topic and avoid unnecessary concepts.

D Sensitivity Analysis

All experiments are performed on a fixed stratified 300-question HotpotQA sample (seed=42). The results for budget sweep and branching factor sweep are depicted in Table 9 and Table 10.

Budget B	5	10	15	20	30
Accuracy (%)	73.0	77.3	78.0	78.7	76.0

Table 9: Budget sweep. Clear monotonic improvement B=5→20 (+5.7 pp), diminishing returns at B=30.

Branching K	1	2	3	4
Accuracy (%)	77.7	78.3	76.0	80.3

Table 10: Branching factor sweep. K=1 (linear chain) degrades gracefully; K=4 achieves the highest accuracy.

E Compute Analysis

Compute–accuracy tradeoff analysis is done in Table 11.

Method	**Acc.%**	**Tok/q**	**Notes**
CoT	76.80	∼512	single call
SC (5 traces)	78.21	∼2,560	5× CoT
A*, K=1, B=30	77.7	3,372	linear chain
A*, K=3, B=15	78.0	12,452	paper default
A*, K=3, B=20	78.7	12,391	peak accuracy

Table 11: Compute–accuracy tradeoff. Token counts are measured from ablation experiments, not estimated.

F Cross-Dataset Topic Modelling

We apply Non-negative Matrix Factorisation (Lee and Seung, 1999) to all pooled QA question texts used for the cross-dataset topic analysis.

Pipeline. (1) Text extraction: question string for StrategyQA/HotpotQA, scenario context for LogicQA (since its query strings are formulaic). (2) TF-IDF: max_df=0.80, min_df=3, max_features=5,000, ngram_range=(1,2), English stop-words removed. (3) NMF: k=8, init=NNDSVD-a, max_iter=500, random_state=42. (4) Topic assignment via argmax. (5) Per-topic A*/CoT accuracy.

Validation. We select k=8 via NPMI coherence (0.301), the smallest k above 0.30 with interpretable topics.

Topic	**Cov.%**	**Str.QA** (A*/CoT)	**Hot.QA** (A*/CoT)	**W**
everyday_reasoning	27.7	79/79	88/73	CoT
entertainment_media	23.9	73/73	83/59	A*
historical_event	12.7	78/78	68/47	A*
geography_location	12.6	70/74	86/55	A*
general_property	9.9	73/72	63/40	A*
hypothetical_scenario	6.1	72/76	100/57	CoT
us_politics_law	4.4	72/81	91/36	A*
biography_origin	2.7	65/70	90/63	A*

Table 12: Per-topic A* vs. CoT accuracy (%) on 14B Qwen-2.5. W = overall winner across datasets.

Key finding. A* wins in 6 of 8 categories covering 66.2% of the corpus (Table 12). All A*-dominant topics share a structural property: they require resolving multi-hop entity or factual chains. The two CoT-dominant categories—everyday_reasoning and hypothetical_scenario—involve commonsense judgements where a single inference step suffices.

Representative examples. *entertainment_media* (A* +10.4 pp): "If you were on a diet, would you have to skip lunch at McDonald's?" requires resolving menu composition plus dietary constraints across multiple facts.

Benchmark	CoT	A*	SC	ToT
HumanEval	67.48	64.42	65.03	71.78
MBPP	39.80	36.80	38.00	40.40
CodeContests	77.40	87.00	99.00	91.20

Table 13: Accuracy (%) on additional code-generation benchmarks using LLaMA-3.1-8B. SC denotes self-consistency and ToT denotes Tree-of-Thought. These results reinforce that no single inference strategy dominates uniformly across tasks.

historical_event (A* +2.4 pp): "Did Harry Houdini's wife make psychics look foolish?" requires chaining Houdini's death → wife's debunking campaign.

geography_location (A* +9.5 pp): "Is Disney associated with Los Angeles County?" requires resolving corporate headquarters location through entity disambiguation.

everyday_reasoning (CoT +0.5 pp): "Is capturing giant squid in natural habitat impossible with no gear?" is answerable in a single commonsense step.

hypothetical_scenario (CoT +2.7 pp): "Were deaths from Apollo 13 eclipsed by other space missions?" requires a single factual comparison.

G Additional Code-Generation Results

To test whether the observed reasoning-strategy trade-off extends beyond question answering and mathematics, we additionally evaluate HumanEval, MBPP, and CodeContests. Table 13 reports the LLaMA-3.1-8B results using the same four inference strategies considered in the main comparison.

The router remains effective across substantially different formats, including boolean QA, multiple-choice logical reasoning, mathematical problems, code generation, and programming. Together with the pooled cluster and topic analyses, these results provide evidence against a purely benchmark-specific explanation. They do not, however, by themselves establish transfer to an entirely unseen benchmark format.

H Reproducibility Details

Models. LLaMA-3.1-8B-Instruct (main backbone), Qwen2-0.5B-Instruct (small-scale comparison), Qwen-2.5-14B-Instruct (scale experiment), DeepSeek-R1-Distill-Qwen-14B (critic). All models served locally via Ollama (v0.3.x) or vLLM (v0.4.x) with greedy decoding for CoT and temperature sampling for A* branches (with temperature of 0.7).

Datasets. StrategyQA: full dev set, 2,290 examples. LogicQA: full test set, 651 examples. HotpotQA: stratified random sample of 500 from the fullwiki dev set (seed=42). GSM8K: test split, 1,319 examples. MATH: test split, 5,000 examples. 14B experiments: overlapping subset where both 8B and 14B produced parseable outputs (243–651 per dataset).

Hardware. All experiments run on a single node with 2× NVIDIA A100 80GB GPUs. Total compute for full experimental pipeline: approximately 340 GPU-hours.

Seeds and variance. CoT uses greedy decoding (deterministic). A* uses temperature-based branching; we report single-run results but the sensitivity analysis (Appendix D) covers hyperparameter variance. Router training uses 5-fold cross-validation with stratified splits (seed=42).

Evaluation sets. StrategyQA contains 2,290 examples, LogicQA contains 651 examples, and we sample 500 examples from the HotpotQA fullwiki development set. Reported paired CoT–A* analyses are computed on the corresponding examples for which the final evaluation pipeline produces valid paired predictions. We use the same paired prediction set for the aggregate CoT/A* accuracies and the instance-level decomposition in Table 2.

I Proof

We prove the result by considering three cases.

Case 1: Root node ($d_n = 0$)**.** By Assumption 1, any valid goal trace requires at least $H_{\min}$ future transitions from the root. Since each future transition costs at least $c_{\min}$, the true remaining cost satisfies

$$h^*(n) \geq H_{\min} c_{\min}.$$

At the root, $d_n = 0$, so

$$h_{\text{depth}}(n) = \omega_d H_{\min}.$$

Also, because $E(n) \in [0, 1]$, we have

$$h_{\text{coverage}}(n) = \omega_c(1 - E(n)) \leq \omega_c.$$

Therefore,

$$h(n) = \omega_d H_{\min} + \omega_c(1 - E(n)) \leq \omega_d H_{\min} + \omega_c.$$

Using $\omega_d + \omega_c \leq c_{\min}$, and hence $\omega_d \leq c_{\min}$, we obtain

Domain	Benchmark	Qwen-0.5B	LLaMA-3.1-8B	Qwen-2.5-14B
QA	StrategyQA	64.74	77.35	76.46
QA	LogicQA	24.98	49.01	58.68
QA	HotpotQA	30.00	82.41	83.17
Math	GSM8K	26.84	91.10	80.82
Math	MATH	33.21	81.52	48.40
Code	HumanEval	34.97	72.50	79.75
Code	MBPP	25.99	40.63	48.42
Code	CodeContests	75.20	95.00	–

Table 14: Topology-router accuracy (%) across benchmarks, domains, and model scales. A dash denotes an unavailable experiment. These experiments test the same 13-feature routing formulation across substantially different input formats.

$$\omega_d H_{\min} + \omega_c = (H_{\min} - 1)\omega_d + (\omega_d + \omega_c) \leq (H_{\min} - 1)c_{\min} + c_{\min} = H_{\min} c_{\min} \leq h^*(n) \tag{8}$$

Thus $h(n) \leq h^*(n)$ at the root.

Case 2: Intermediate non-goal node $(0 < d_n < H_{\min})$**.** Let

$$m = H_{\min} - d_n.$$

Since $0 < d_n < H_{\min}$, we have $m \geq 1$. By Assumption 1, at least m further transitions are required before reaching a goal. Since each transition costs at least $c_{\min}$,

$$h^*(n) \geq m c_{\min}.$$

For such a node,

$$h_{\text{depth}}(n) = \omega_d m.$$

Again, since $E(n) \in [0, 1]$,

$$h_{\text{coverage}}(n) = \omega_c(1 - E(n)) \leq \omega_c.$$

Therefore,

$$h(n) = \omega_d m + \omega_c(1 - E(n)) \leq \omega_d m + \omega_c.$$

Using $\omega_d + \omega_c \leq c_{\min}$ and $\omega_d \leq c_{\min}$, we get

$$\omega_d m + \omega_c = (m-1)\omega_d + (\omega_d + \omega_c) \leq (m-1)c_{\min} + c_{\min} = m c_{\min} \leq h^*(n). \tag{9}$$

Thus $h(n) \leq h^*(n)$ for all intermediate non-goal nodes.

Case 3: Deep non-goal node $(d_n \geq H_{\min})$**.** By Assumption 2, even though the node has already reached the minimum depth, any non-goal node still requires at least one additional transition before reaching a goal. Hence,

$$h^*(n) \geq c_{\min}.$$

For $d_n \geq H_{\min}$, the depth term is zero:

$$h_{\text{depth}}(n) = \omega_d \max(0, H_{\min} - d_n) = 0.$$

Thus,

$$h(n) = \omega_c(1 - E(n)).$$

Since $E(n) \in [0, 1]$,

$$h(n) \leq \omega_c.$$

Because $\omega_c \leq \omega_d + \omega_c \leq c_{\min}$, we have

$$h(n) \leq c_{\min} \leq h^*(n).$$

Thus $h(n) \leq h^*(n)$ also holds for deep non-goal nodes.

Since the inequality $h(n) \leq h^*(n)$ holds in all three cases, the heuristic is admissible for all non-goal nodes under Assumptions 1 and 2. □

Interpretation. The theorem should be read as a conservative guarantee for the depth-plus-coverage heuristic under explicit assumptions. It does not claim that the heuristic is optimal, universally best, or valid for every dataset-specific implementation. In particular, the theorem does not cover the LogicQA-specific remaining-depth/logical-complexity heuristic. Its role is narrower: it shows that the depth-plus-coverage heuristic used in StrategyQA and HotpotQA can be interpreted as a conservative lower bound on the remaining path cost when the stated assumptions and weight condition hold. The empirical robustness of the overall A* procedure is evaluated separately through sensitivity analysis over node budget and branching factor.

J Discussion

The CoT/A* boundary is structural, not dataset-specific. The strongest empirical finding is that A* advantage tracks branching coefficient across clusters (-0.4 pp in C1 to +11.4 pp in C4, Table 3), not benchmark labels. This means the question "should I use search?" is better answered by looking at the input's structure than by knowing which dataset it comes from.

Pre-generation routing beats trace inspection. This was the most surprising result to us. The topology router sees zero generated tokens yet outperforms critics that read both full traces. The explanation is the Fluency-Correctness Asymmetry: in 38% of disagreements, the wrong answer comes wrapped in the more fluent trace, fooling any judge that conflates readability with correctness. The implication is that adaptive inference should start with structural triage, not trace evaluation. The critic-based heuristic itself operates conservatively (mean 0.43 vs. human 0.58, $\epsilon_{\max} = 0.14$), providing practical confidence without formal admissibility.

Substantial room remains. The topology router recovers only 26.1% of the oracle gap for StrategyQA. The remaining likely involves factors we do not yet model: whether the base LLM has the right parametric knowledge for a given question, whether the search depth is matched to the problem, and how model capacity interacts with topology (the 14B results in Table 1 hint at complex interactions).