Multi-chain Graph Refinement and Selection for Reliable Reasoning in Large Language Models

Anonymous ACL submission

Abstract

Complex reasoning remains a critical bottleneck for the practical deployment of Large Language Models (LLMs). Test-time expansion methods such as Tree-of-Thought (ToT) and Graph-of-Thought (GoT) improve reasoning by exploring intermediate structures via tree search or graph-based mechanisms. However, they operate within a single homogeneous structure, which leaves their exploration short on diversity, wastes computation on redundant branches, and offers no principled way to integrate or cross-correct genuinely different lines of reasoning. We propose Multi-chain Graph Refinement & Selection (MGRS), a reasoning framework that replaces such elaborate search structures with a small set of differentiated branches, each following a distinct line of reasoning to retain search exploration at far lower cost. Built around these differentiated branches, MGRS extends self-verification with inter-branch cross-verification to validate and refine each reasoning chain, constructs a reasoning relation graph through intra-chain dependency analysis and cross-chain node merging, and ultimately selects the most reliable answer and its reasoning trajectory via node-level confidence estimation and Noisy-OR aggregation. Across eight benchmarks spanning four task types, MGRS attains an average accuracy of 82.6%, outperforming the strongest baseline by 2.5 points while invoking fewer LLM calls. On two search-intensive puzzles, MGRS is the first to reach 100% accuracy on Game of 24, with a 13.6× speed-up over the leading Forest-of-Thought (FoT) framework; on the harder Countdown, it surpasses FoT by 23 points (88.4% vs. 65.3%) while running 2.3× faster.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a broad range of language tasks, yet complex multi-step reasoning remains a persistent challenge (Lin et al., 2025). The conventional next-token prediction mechanism is prone to systematic biases and error accumulation, causing performance to degrade as reasoning chains lengthen. The Chain-of-Thought (CoT) framework (Wei et al., 2022) alleviates this by augmenting prompts with explicit intermediate reasoning steps, guiding models toward structured multi-step inference and yielding substantial gains on mathematical, commonsense, and symbolic reasoning benchmarks. Yet a single CoT chain remains vulnerable to early errors that propagate irreversibly, motivating the development of richer test-time reasoning frameworks.

A prominent line of work extends CoT by organizing the reasoning process as a tree or graph structure (Yao et al., 2023; Besta et al., 2024; Bi et al., 2025), enabling multi-branch exploration and global search over intermediate reasoning states. By maintaining multiple candidate solutions simultaneously and using heuristic evaluation or beam search to guide branch expansion, these methods substantially extend the reasoning horizon of LLMs. However, existing methods typically instantiate branches by repeatedly expanding a largely shared reasoning paradigm: branches may explore different intermediate states, but do not differ substantially in the underlying reasoning strategy. This leaves their exploration short on genuine diversity, wastes computation on near-redundant branches, and offers no principled mechanism to cross-validate or correct one reasoning path using insights from a fundamentally different one.

To address this fundamental limitation, we propose Multi-chain Graph Refinement & Selection (MGRS), a reasoning framework that replaces homogeneous branch exploration with a compact set of **differentiated branches**, each pursuing a structurally distinct reasoning strategy such as problem contraction and backward reasoning. Built on this differentiated foundation, MGRS performs inter-branch self- and cross-verification to mutu-

ally validate and refine each reasoning chain, constructs a reasoning relation graph through intra-chain dependency analysis and cross-chain node merging, and selects the most reliable answer via node-level success-rate estimation and Noisy-OR aggregation. Across eight benchmarks spanning four task types, MGRS attains an average accuracy of 82.6%, outperforming the strongest baseline by 2.5 points while invoking fewer LLM calls. On search-intensive tasks, MGRS is the first to achieve 100% accuracy on Game of 24 with a 13.6× speed-up over the leading Forest-of-Thought framework, and surpasses it by 23 points on the harder Countdown benchmark while running 2.3× faster.

Our main contributions are as follows:

- We propose MGRS, a multi-chain graph refinement and selection framework that replaces homogeneous tree or graph expansion with a compact set of differentiated reasoning branches, preserving exploratory capacity while reducing redundant computation.

- We design a composite refinement and selection mechanism that introduces self- and cross-verification, reasoning-relation graph construction, and node-level confidence aggregation as coordinated stages for identifying the most reliable answer and its supporting reasoning trajectory.

- Extensive experiments across eight benchmarks and two search-intensive arithmetic puzzles show that MGRS consistently improves reasoning accuracy while requiring fewer LLM invocations than strong reasoning baselines.

2 Related Work

2.1 Structured reasoning optimization

Structured reasoning methods improve LLM inference by making intermediate reasoning states explicit. Chain-of-Thought prompting (CoT) (Wei et al., 2022) elicits step-by-step solutions and has been extended by zero-shot prompting (Kojima et al., 2022), planning-based decomposition (Wang et al., 2023a), and memory or backtracking mechanisms for longer reasoning chains (Luo et al., 2025). Despite their effectiveness, chain-based methods remain inherently linear: early mistakes can propagate through the entire trajectory, and a single reasoning strategy provides limited coverage of alternative solution structures.

To expand the reasoning space, tree- and graph-based frameworks organize intermediate states into richer search structures. Tree of Thoughts (ToT) (Yao et al., 2023; Long, 2023) explores multiple candidate continuations with search and self-evaluation, while later methods incorporate Monte Carlo Tree Search (Zhang et al., 2024; Kocsis and Szepesvári, 2006), input augmentation, or multi-sampling strategies (Ding et al., 2024; Bi et al., 2025). Graph of Thoughts (GoT) (Besta et al., 2024) generalizes this idea by allowing reasoning states to be merged, skipped, or revisited in a graph topology, and related works further investigate graph-structured reasoning operators (Yao et al., 2024). Atom of Thoughts (AoT) (Teng et al., 2025) decomposes reasoning into atomic units and models their dependencies as a directed graph. In a complementary direction, workflow-based methods such as AFlow (Zhang et al., 2025b) automate the generation of agent reasoning workflows, and multi-agent architecture search approaches such as MaAS (Zhang et al., 2025a) learn to select and compose operators across agents, extending structured reasoning to multi-step agentic settings.

These methods improve exploration, but most of them still rely on homogeneous branches generated under the same reasoning paradigm. As a result, additional branches may be redundant, and the framework lacks an explicit mechanism for using one qualitatively different reasoning path to verify or repair another. MGRS addresses this limitation by replacing large homogeneous search structures with a compact set of differentiated reasoning chains, then constructing a merged reasoning graph from their verified dependencies.

2.2 Controlled reasoning optimization

Another line of work improves test-time reasoning through sampling, aggregation, and verification. Self-consistency and related multi-sampling methods generate multiple reasoning paths and aggregate answers by majority voting or confidence-based criteria (Wang et al., 2023b; Liang et al., 2024; Zhou et al., 2024). Recent work further explores path reranking and answer aggregation to select more reliable outputs from diverse candidates (Chu et al., 2024; Jiang et al., 2025). These approaches reduce variance but typically sample from the same underlying reasoning distribution, which limits the diversity of candidate reasoning

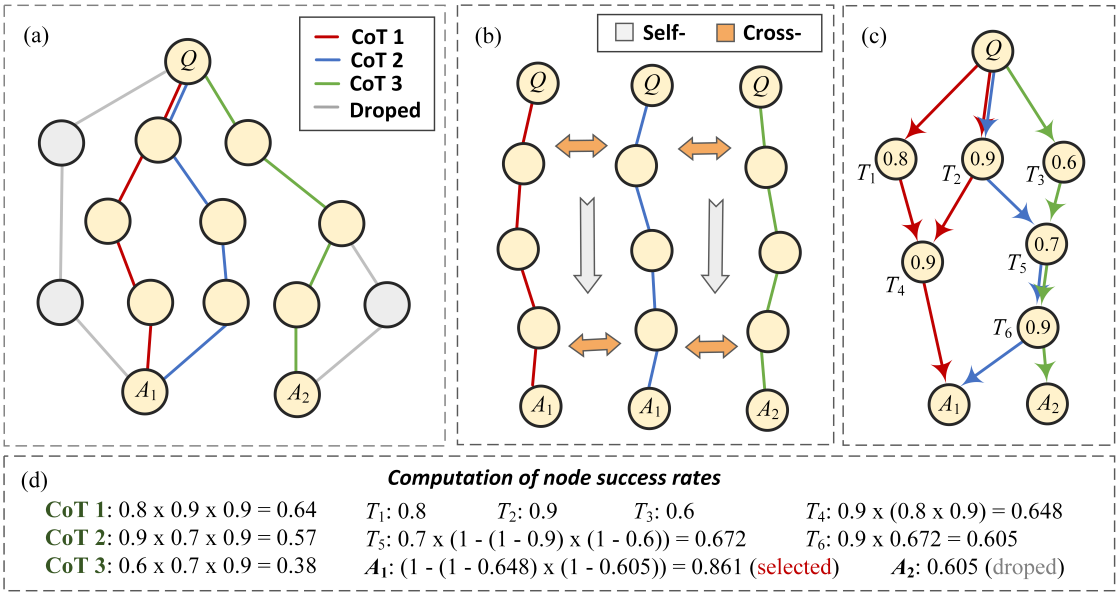


Figure 1: Overview of MGRS. Given a question, MGRS generates differentiated reasoning chains, refines them through self- and cross-verification, constructs a reasoning-relation graph by dependency analysis and node merging, and selects the final answer by confidence-based aggregation.

paths and provides no mechanism to cross-validate across fundamentally different strategies.

Verification-based methods aim to detect and correct reasoning errors during or after generation. Representative approaches include post-hoc verify-and-edit strategies (Zhao et al., 2023), deductive step-level checking (Ling et al., 2023), iterative self-refinement (Madaan et al., 2023), and tool-interactive self-correction such as CRITIC (Gou et al., 2024), which grounds revision in executable feedback from external tools. Verification can be performed internally by the model or externally with tools, knowledge sources, or discriminators, improving the reliability of individual reasoning chains.

MGRS builds on these ideas but adapts them to a multi-branch setting. Instead of only verifying each chain in isolation or aggregating final answers, MGRS performs both intra-chain self-verification and inter-chain cross-verification, allowing differentiated branches to validate and refine one another. It then estimates node-level confidence in the resulting reasoning graph and applies Noisy-OR aggregation to select the most reliable answer and reasoning trajectory.

3 Methodology

We describe MGRS in four stages, following the overview in Figure 1: differentiated chain generation, composite verification and refinement, reasoning-relation graph construction, and confidence-based answer selection.

3.1 Differentiated reasoning chains generation

MGRS first generates a compact set of differentiated reasoning chains $\Gamma = \{T^{(1)}, T^{(2)}, \ldots, T^{(M)}\}$ for each question Q. Each chain is guided by a prompt $P^{(i)}$ that combines a standard multi-step reasoning instruction with a branch-specific strategy. We write its sampling process compactly as:

$$A^{(i)} \sim p\left(A, T^{(i)}_{0:N} \mid Q, P^{(i)}\right). \quad (1)$$

Here, $A^{(i)}$ is the answer produced by the i-th chain, and $T^{(i)}_j$ is its j-th reasoning step, generated autoregressively from previous steps, the question, and the branch prompt. The differentiated component in $P^{(i)}$ can be derived from prior-method summaries, automatically generated heuristics, or predefined theoretical guidance, encouraging the model to explore distinct reasoning perspectives.

To improve stability, the first m branches ($1 < m < M$) are sampled N times independently. For each sampled chain, we compute a perplexity-based confidence score over its final-answer tokens $\{x_l\}_{l=1}^{L}$:

$$S^{(k)} = \exp\left(-\frac{1}{L}\sum_{l=1}^{L}\log p(x_l \mid x_{<l})\right). \quad (2)$$

A lower $S^{(k)}$ indicates higher model confidence. For each branch, we retain the top-R chains with the lowest scores, preserving high-confidence intra-branch variants for subsequent refinement. Differentiated chains may include both forward and reverse trajectories, with reverse reasoning further discussed in Section 4.2.

3.2 Composite verification and refinement

Building on self-correction methods such as Self-Refine (Madaan et al., 2023), MGRS extends verification to a multi-branch setting through intra-chain self-verification and inter-chain cross-verification. In self-verification, the LLM checks each chain step by step for logical or computational errors, optionally assisted by lightweight rule-based checks such as arithmetic validation. When an error is found, the flawed segment is corrected and the downstream reasoning is regenerated conditioned on the revision.

Cross-verification then compares the refined outcomes across different branches. If branches disagree, MGRS traces the divergent segments and applies the same refinement procedure to the corresponding parts. This composite mechanism allows differentiated branches to validate and repair one another, improving both accuracy and robustness.

3.3 Reasoning relation graph construction

After verification, MGRS converts the refined chains into a reasoning-relation graph through two operations: intra-chain dependency analysis and cross-chain node merging. The first operation identifies step-to-step dependencies within each reasoning chain and forms chain-specific subgraphs. The second merges semantically equivalent sub-steps across different branches, thereby integrating these subgraphs into a unified directed acyclic graph (DAG).

The LLM then assigns an estimated success probability to each merged node, and duplicate dependencies are removed after edge unification. The resulting graph is defined as:

$$G = (S, E)\,, \quad S = \{S_i\}_{i=1}^{n}\,, \quad E \subseteq S \times S \quad (3)$$

Each node S_i represents a merged sub-step with weight W_i, where $0 < W_i < 1$ denotes the estimated success rate. Each edge (S_i, S_j) indicates that S_j provides information required by S_i.

3.4 Answer and reasoning strategy selection

After constructing the reasoning graph, MGRS scores candidate answers by propagating node-level success rates through the DAG. For a linear chain, all steps must be correct, so the chain success rate is:

$$P_{\text{CoT}} = \prod_{i=1}^{n} W_i. \quad (4)$$

For the reasoning graph, cumulative success is propagated in topological order. Source nodes use $P(S_i) = W_i$, and single-parent nodes use $P(S_i) = W_i P(S_j)$. When a node has parents from different chains, these parents provide alternative supports; therefore, we use a Noisy-OR aggregation rule:

$$P(S_i) = W_i \times \left[1 - \prod_{k=1}^{n} (1 - P(S_{jk}))\right]. \quad (5)$$

If multiple parents come from the same chain, as illustrated in Figure 1(c), they represent sequential rather than alternative support. We first merge them with the product rule in Equation 4 and then apply Equation 5.

By applying these rules in topological order, MGRS obtains the cumulative success rate of each answer node and selects the answer together with its highest-scoring reasoning trajectory.

4 Experiments

We evaluate MGRS along four dimensions: general reasoning performance, search-intensive arithmetic reasoning, reliability and scaling behavior, and component-level ablations. Together, these experiments examine whether MGRS improves accuracy, reduces test-time computation, and remains robust across settings.

4.1 Main Experiments

Benchmarks and metrics. We evaluate MGRS on eight widely used benchmarks spanning five categories of reasoning tasks: mathematical reasoning with MATH (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021), logical reasoning with the multiple-choice subset of BBH (Suzgun et al., 2023), knowledge-intensive reasoning with MMLU-CF (Zhao et al., 2025), multi-hop reasoning with HotpotQA (Yang et al., 2018) and LongBench (Bai et al., 2024), and code reasoning with MBPP and HumanEval+. We report exact-match accuracy for MATH, GSM8K, BBH, MMLU-CF,

Table 1: Performance comparison across eight reasoning benchmarks.

Method	MATH	GSM8K	BBH	MMLU-CF	HotpotQA	LongBench	MBPP	HEval+	Avg.	Calls/q
CoT	78.3	90.9	78.3	69.6	67.2	57.6	72.4	78.0	74.0	1.0
CoT-SC (n=5)	81.8	92.0	83.4	71.1	66.2	58.6	73.2	79.6	75.7	5.0
Self-refine	78.7	91.7	80.0	69.7	68.3	58.2	72.8	82.7	75.3	3.0
CRITIC	81.9	92.7	78.9	70.3	65.8	58.9	74.1	82.1	75.6	3.0
AFlow	83.0	93.5	76.0	69.5	73.5	61.0	74.0	83.6	76.8	6.5
MaAS	82.9	92.3	82.7	68.0	69.4	64.1	76.0	81.8	77.2	5.0
FoT (n=8)	82.5	94.0	82.4	70.6	66.7	59.1	74.8	83.5	76.7	>8.0
AoT	83.6	95.0	86.0	70.9	80.6	68.5	75.2	80.9	80.1	7.7
MGRS (Ours)	**86.8**	**96.5**	**86.2**	**74.2**	**83.5**	**69.9**	**78.3**	**85.7**	**82.6**	**6.1**

Table 2: Cross-model generalization across four representative benchmarks and three backbone LLMs.

	MaAS				**AoT**				**MGRS (Ours)**			
Dataset	Qwen	DS-V3	Llama	Avg.	Qwen	DS-V3	Llama	Avg.	Qwen	DS-V3	Llama	Avg.
MATH	96.0	95.2	96.5	95.9	95.6	94.7	97.1	95.8	98.2	96.1	98.6	**97.6**
BBH	86.9	84.7	94.5	88.7	89.1	85.6	98.3	91.0	91.4	88.3	98.8	**92.8**
HotpotQA	71.8	73.4	75.5	73.6	73.5	74.9	77.3	75.2	75.5	77.7	80.1	**77.8**
HEval+	94.0	91.5	74.0	86.5	93.2	92.0	72.2	85.8	97.5	95.1	78.0	**90.2**

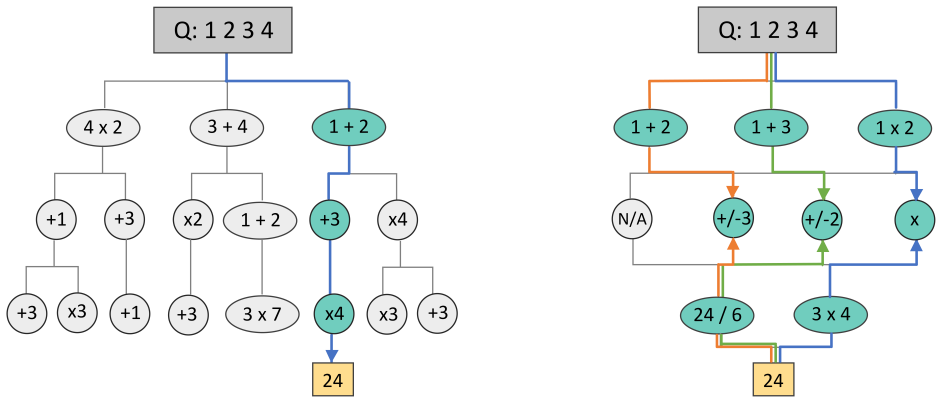
and MBPP, F1 scores for HotpotQA and LongBench, and pass@1 for HumanEval+. We also report the average number of LLM invocations per question for efficiency comparison.

Baselines. We compare MGRS against representative methods from three lines of test-time reasoning research: prompting and refinement methods, including CoT (Wei et al., 2022), CoT-SC (Wang et al., 2023b), Self-Refine (Madaan et al., 2023), and CRITIC (Gou et al., 2024); workflow- and agent-based methods, including AFlow (Zhang et al., 2025b) and MaAS (Zhang et al., 2025a); and structured reasoning frameworks, including FoT (Bi et al., 2025) and AoT (Teng et al., 2025). All methods use the same backbone model when applicable.

Evaluation setting. Following the setting of AoT (Teng et al., 2025), we use the full GSM8K test set with 1,319 samples, the merged MuSiQue (Trivedi et al., 2022) and WikiMultiHopQA (Ho et al., 2020) subsets for LongBench with 400 samples, and the first 1,000 test samples for the remaining non-code benchmarks when applicable. The main benchmark experiments use GPT-4o-mini as the backbone model to balance reasoning capability and inference cost. Unless otherwise specified, MGRS uses two differentiated reasoning branches and four samples per branch.

Overall results. Table 1 reports the main results across the eight benchmarks. MGRS achieves the best average performance, reaching 82.6%, which improves over the strongest baseline AoT by 2.5 points while using fewer LLM calls on average. The improvements are broad across task types. On mathematical reasoning, MGRS outperforms AoT by 3.2 points on MATH and 1.5 points on GSM8K. On knowledge-intensive and multi-hop reasoning, it improves over AoT by 3.3 points on MMLU-CF, 2.9 points on HotpotQA, and 1.4 points on LongBench, while remaining competitive on BBH. MGRS also improves code reasoning performance, surpassing AoT by 3.1 points on MBPP and 4.8 points on HumanEval+. Beyond accuracy, MGRS requires only 6.1 LLM calls per question on average, compared with 7.7 for AoT and more than 8.0 for FoT, suggesting that differentiated branching and cross-chain refinement improve the efficiency-accuracy trade-off rather than merely increasing test-time computation.

Cross-model generalization. To examine whether the gains of MGRS depend on a specific backbone, we further evaluate it on Qwen3.5-27B, DeepSeek-V3, and Llama-3.1-70B across four representative tasks: MATH, BBH, HotpotQA, and HumanEval+. As shown in Table 2, MGRS achieves the best average performance on all four tasks and attains the highest score in every model–task setting. Averaged over the three backbones, MGRS reaches 97.6% on MATH, 92.8% on BBH, 77.8% on HotpotQA, and 90.2% on HumanEval+, yielding an overall mean of 89.6% across the four tasks. This corresponds to an average improvement of 2.6 points over AoT and 3.4 points over MaAS, with the largest gains on HumanEval+ (+4.4) and



(a) Unidirectional reasoning tree of FoT

(b) Bidirectional reasoning tree of MGRS

Figure 2: Comparison between the reasoning tree of FoT and the bidirectional reasoning structure of MGRS for search-intensive arithmetic reasoning. MGRS uses forward and backward reasoning branches to search from the input numbers and the target value, respectively.

Table 3: Performance comparison on Game of 24 and Countdown. "Succ." denotes success rate.

Method	Game of 24		Countdown	
	Calls	Succ.	Calls	Succ.
CoT	1.0	4.4	1.0	5.3
CoT-SC	10.0	4.4	10.0	6.3
AoT	4.0	11.6	4.0	10.5
GoT (k=1)	7.0	5.3	8.6	5.3
ToT	13.7	74.0	13.1	33.7
BoT	15.0	82.4	21.9	44.2
XoT	19.0	86.3	30.8	49.4
FoT (n=4)	20.6	93.7	31.2	65.3
MGRS (Ours)	**7.3**	**100.0**	**17.6**	**88.4**

Table 4: Runtime comparison on Game of 24 and Countdown. Runtime is measured in hours.

Task	BoT	XoT	FoT	MGRS
Game of 24	7.7	10.1	12.2	**0.9**
Countdown	9.9	14.1	13.9	**6.1**

HotpotQA (+2.6). These results suggest that the benefits of differentiated reasoning, cross-chain refinement, and graph-based selection are not tied to a particular LLM backbone.

4.2 Search-Intensive Arithmetic Reasoning

Tasks and metrics. We further evaluate MGRS on Game of 24 and Countdown, two search-intensive arithmetic benchmarks that require exploring and composing intermediate arithmetic states. These tasks are designed to test whether differentiated reasoning branches can reduce homogeneous tree expansion in combinatorial search. Following FoT (Bi et al., 2025), we use the same 95 valid Game of 24 cases and sample 95 Countdown puzzles. We report success rate, average LLM calls, and normalized runtime.

Baselines. We compare MGRS with prompting-based methods, including CoT (Wei et al., 2022) and CoT-SC (Wang et al., 2023b); structured reasoning methods, including AoT (Teng et al., 2025), ToT (Yao et al., 2023), and GoT (Besta et al., 2024); and stronger tree-search variants, including BoT (Yang et al., 2024), XoT (Ding et al., 2024), and FoT (Bi et al., 2025).

Evaluation setting. All experiments in this section use Mistral-7B-Instruct as the backbone model. The FoT baseline follows a breadth-first search implementation with a branching factor of four ($n = 4$). Search-based baselines follow the FoT evaluation protocol when applicable; otherwise, we use the recommended settings from the original papers. For MGRS, we initialize two differentiated branches that solve the problem from opposite directions: a forward branch starting from the given numbers and a backward branch starting from the target value, as illustrated in Figure 2. If the two branches converge at the same intermediate state, the solution is obtained immediately; otherwise, MGRS continues with the standard forward reasoning process. Convergence and final expressions are checked by the deterministic arithmetic verifier, which requires no additional LLM calls.

Overall results. Table 3 shows that MGRS achieves the highest success rate on both arithmetic search tasks while requiring fewer LLM calls than strong tree-search baselines. On Game of 24, MGRS reaches 100.0% success in our evaluated setting, improving over FoT by 6.3 points

while reducing the average number of LLM calls from 20.6 to 7.3. The advantage becomes more pronounced on the harder Countdown task, where MGRS reaches 88.4% success, improving over FoT by 23.1 points while reducing calls from 31.2 to 17.6. In contrast, methods with limited explicit search in this setting, such as CoT, CoT-SC, AoT, and GoT, remain below 12% success on both tasks, suggesting that explicit combinatorial search remains necessary for these puzzles. These results indicate a stronger accuracy–cost trade-off rather than more test-time computation.

Runtime analysis. Table 4 further shows that MGRS substantially reduces wall-clock runtime compared with other multi-branch reasoning methods. On Game of 24, MGRS completes the evaluation in 0.9 hours, corresponding to a $13.6\times$ speedup over FoT. On Countdown, it completes the evaluation in 6.1 hours, yielding a $2.3\times$ speedup. This efficiency gain comes from the bidirectional design of MGRS. In standard one-directional tree search, the number of candidate states grows rapidly as the search depth increases. By introducing a backward branch from the target side, MGRS reduces the depth that must be explored in a purely forward direction. When the forward and backward branches meet at a valid intermediate state, MGRS can terminate early without expanding the remaining forward search space.

4.3 Analysis of Reliability and Scaling

Success-Rate Calibration. Since MGRS scores reasoning paths using LLM-estimated step-level success rates (SR), we test whether they provide reliable selection signals. We pair each predicted SR value with a binary step-correctness label, obtained from task-specific automatic checks when available and otherwise from consistency with the final correct reasoning trajectory. We then apply 5-fold cross-validated Platt scaling and report Pearson correlation, expected calibration error (ECE), Brier score, AUROC, and the SR gap between correct and incorrect steps in Table 5.

As shown in Table 5, ECE remains consistently low across all datasets (≤ 0.018), indicating low calibration error. AUROC is above the random baseline of 0.5 on all evaluated datasets, ranging from 0.561 on HotpotQA to 0.636 on MATH, and the SR gap between correct and incorrect steps is always positive. The gap is largest on MATH ($+0.162$) and remains positive even on the more challenging HumanEval+ setting ($+0.028$), where

Table 5: Calibration of MGRS-predicted step-level success rates.

Dataset	Pearson r	ECE	Brier	AUROC	SR(c)	SR(i)	Gap
BBH	0.278	0.018	0.160	0.608	0.795	0.715	+0.081
MATH	0.402	0.016	0.140	0.636	0.822	0.661	+0.162
HotpotQA	0.202	0.005	0.224	0.561	0.643	0.602	+0.041
HumanEval+	0.168	0.001	0.107	0.572	0.877	0.849	+0.028

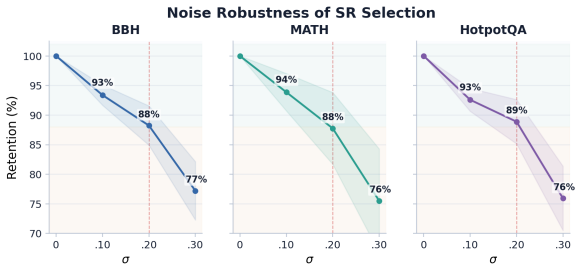


Figure 3: Robustness of the SR discrimination gap to Gaussian noise. Shaded regions indicate 95% confidence intervals over 50 perturbation trials.

local reasoning traces are less diagnostic of final code correctness. These results indicate that SR estimates contain useful discriminative signal for confidence-based path selection.

Robustness to SR Noise. We next test whether path selection is sensitive to imperfect SR estimates by injecting Gaussian noise with $\sigma \in \{0.10, 0.20, 0.30\}$ into the SR values. Figure 3 reports the discrimination gap between correct and incorrect reasoning steps over 50 trials.

As shown in Figure 3, across BBH, MATH, and HotpotQA, the gap decreases smoothly as noise increases but remains positive under all tested perturbations. The 95% confidence intervals remain relatively tight across all noise levels, suggesting stable behavior across perturbation trials. On average, the gap retains 88% of its noise-free value at $\sigma = 0.20$ and 76% at $\sigma = 0.30$. This suggests that MGRS does not rely on perfectly calibrated absolute SR values; the relative ordering of node confidence remains sufficiently stable for Noisy-OR-based path selection.

Chain-Scaling Analysis. We next study how MGRS scales with differentiated reasoning chains, and compare it with scaling CoT-SC by drawing more samples from the same prompting distribution. We sweep MGRS chains $n_b \in \{1, 2, 4\}$ and CoT-SC samples $n \in \{1, 3, 5, 8\}$ on BBH, MATH, and HotpotQA under the same random seed. Table 6 reports performance and average LLM calls.

Table 6 shows that single-chain MGRS already outperforms single-sample CoT-SC on all three datasets, suggesting that the gains do not come solely from chain ensembling. As the test-time

Table 6: Chain-scaling comparison between MGRS and CoT-SC. n_b and n denote MGRS chains and CoT-SC samples, respectively, with n_s=1.

Method	BBH		MATH		HotpotQA	
	Acc.	Calls	Acc.	Calls	Acc.	Calls
CoT-SC n=1	75.0	1.0	73.0	1.0	66.8	1.0
CoT-SC n=3	79.0	3.0	78.0	3.0	71.7	3.0
CoT-SC n=5	76.0	5.0	76.0	5.0	73.9	5.0
CoT-SC n=8	80.0	8.0	80.0	8.0	75.3	8.0
MGRS n_b=1	76.8	2.4	81.4	2.1	79.6	2.8
MGRS n_b=2	82.0	3.9	83.1	3.6	80.5	4.6
MGRS n_b=4	**86.2**	6.8	**86.6**	5.3	**83.3**	6.9

Table 7: Ablation study on core MGRS components.

Method	BBH	MMLU-CF
MGRS (Full)	85.8	74.7
- Success Rate Estimation	84.6	72.9
- Self- & Cross-Verification	84.8	73.0
- Graph Construction	84.3	73.6

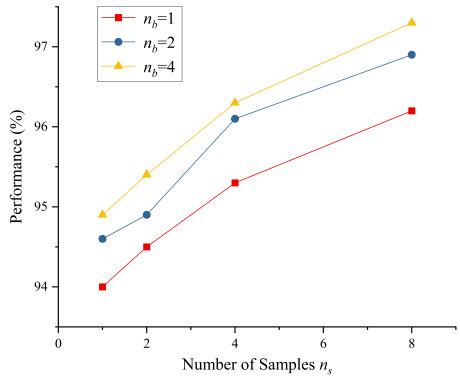


Figure 4: Effect of the number of reasoning branches and samples per branch on MGRS performance.

budget increases, MGRS with four differentiated chains exceeds CoT-SC with eight samples while using fewer calls, indicating a better accuracy–cost trade-off. CoT-SC shows early saturation as additional samples are drawn from the same reasoning distribution, whereas MGRS continues to benefit from structurally differentiated chains. This supports the design choice of MGRS: increasing strategic diversity across chains can be more effective than merely increasing homogeneous samples.

4.4 Ablation Study

We conduct two ablation studies to isolate the contribution of MGRS's key design choices. The first removes core modules of MGRS on BBH and MMLU-CF to evaluate their individual effects. The second studies how performance changes with the number of differentiated reasoning branches and samples per branch on GSM8K.

Component ablation. Table 7 shows that removing any core component degrades performance. Success-rate estimation contributes 1.2–1.8 points, self- and cross-verification contributes 1.0–1.7 points, and graph construction contributes 1.1–1.5 points across BBH and MMLU-CF. These drops confirm that confidence-aware selection, verification-based refinement, and graph-structured consolidation all contribute to the final performance of MGRS.

Branch and sample scaling. We vary the number of reasoning branches ($n_b \in \{1, 2, 4\}$) and the number of samples per branch ($n_s \in \{1, 2, 4, 8\}$). As shown in Figure 4, increasing either dimension generally improves performance, with the best accuracy of 97.3% achieved at $n_b = 4$ and $n_s = 8$. Increasing n_s improves generation stability but shows diminishing returns at larger sampling budgets. Increasing n_b also improves performance, especially from one to two branches, confirming the value of differentiated reasoning. The smaller marginal gain beyond two branches suggests that excessive differentiation may be less beneficial on GSM8K, whose solution space is relatively constrained.

5 Conclusion

This paper introduces Multi-chain Graph Refinement and Selection (MGRS), a test-time reasoning framework that replaces large homogeneous tree- or graph-based expansion with a compact set of differentiated reasoning branches. By combining self- and cross-verification, reasoning-relation graph construction, and confidence-based answer selection, MGRS improves reasoning reliability while reducing redundant computation. Experiments across eight general reasoning benchmarks and two search-intensive arithmetic puzzles show that MGRS consistently improves the accuracy–cost trade-off over strong reasoning baselines.

6 Limitations

A limitation of MGRS is that it currently uses task-informed prompts to instantiate differentiated reasoning branches. While this design is effective across the evaluated tasks, choosing the most useful branch types may require task-specific knowledge. For tasks with highly standardized solution procedures, adding additional differentiated branches may bring limited marginal benefit. Automating branch discovery and prompt optimization is a promising direction for reducing manual tuning and improving adaptability across task domains and a wider range of reasoning problems.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2025. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*.

Zheng Chu, Jingchang Chen, Qianglong Chen, Haotian Wang, Kun Zhu, Xiyuan Du, Weijiang Yu, Ming Liu, and Bing Qin. 2024. Beamaggr: Beam aggregation reasoning over multi-source knowledge for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2024. Everything of thoughts: Defying the law of penrose triangle for thought generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1638–1662.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems*, volume 34.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Eric Hanchen Jiang, Haozheng Luo, Shengyuan Pang, Xiaomin Li, Zhenting Qi, Hengli Li, Cheng-Fu Yang, Zongyu Lin, Xinfeng Li, Hao Xu, and 1 others. 2025. Learning to rank chain-of-thought: An energy-based approach with outcome supervision. *arXiv preprint arXiv:2505.14999*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, and 1 others. 2024. Internal consistency and self-feedback in large language models: A survey. *arXiv preprint arXiv:2407.14507*.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. 2025. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. In *Proceedings of the 42nd International Conference on Machine Learning*.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36:36407–36433.

Jieyi Long. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Hongyin Luo, Nathaniel Morgan, Tina Li, Derek Zhao, Ai Vy Ngo, Philip Schroeder, Lijie Yang, Assaf Ben-Kish, Jack O'Brien, and James Glass. 2025. Beyond context limits: Subconscious threads for long-horizon reasoning. *arXiv preprint arXiv:2507.16784*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Fengwei Teng, Quan Shi, Zhaoyang Yu, Jiayi Zhang, Yuyu Luo, Chenglin Wu, and Zhijiang Guo. 2025. Atom of thoughts for markov llm test-time scaling. In *Advances in Neural Information Processing Systems*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. 2024. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544.

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.

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. *Advances in neural information processing systems*, 36:11809–11822.

Yao Yao, Zuchao Li, and Hai Zhao. 2024. Got: Effective graph-of-thought reasoning in language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2901–2921.

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. *arXiv preprint arXiv:2406.07394*.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. 2025a. Multi-agent architecture search via agentic supernet. In *Proceedings of the 42nd International Conference on Machine Learning*. Spotlight.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2025b. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*.

Qihao Zhao, Yangyu Huang, Tengchao Lv, Lei Cui, Qinzheng Sun, Shaoguang Mao, Xin Zhang, Ying Xin, Qiufeng Yin, Scarlett Li, and 1 others. 2025. Mmlu-cf: A contamination-free multi-task language understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13371–13391.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Wenjie Zhou, Qiang Wang, Mingzhou Xu, Ming Chen, and Xiangyu Duan. 2024. Revisiting the self-consistency challenges in multi-choice question formats for large language model evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14103–14110.