Reasoning / Large Language Models

Mutual Reasoning Makes Smaller LLMs Stronger Problem-Solvers (rStar)

Authors: Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, Mao Yang.

Affiliation: Microsoft Research Asia (MSRA) and Harvard University

Publication: ICLR 2025

Year: 2025

1. Summary

Problem

What problem does the paper address?

- Large Language Models (LLMs) still struggle with **complex multi-step reasoning**, especially on mathematical and logical tasks.
- This problem is **more severe for Small Language Models (SLMs)**, which often explore ineffective reasoning paths and cannot reliably evaluate or verify their own reasoning.
- Existing Solutions require expensive fine-tuning.

Why is it important?

- Improving SLM reasoning **without fine-tuning or stronger teacher models** reduces training cost and dependence on proprietary LLMs.
- It enables SLMs to perform **more accurate reasoning at inference time**, making them more practical for real-world and resource-constrained applications.

Method

Core idea

- Uses Monte Carlo Tree Search (MCTS) augmented with SLM to generate many reasoning trajectories.
- Introduces five human-like reasoning actions.
- Uses a second SLM to verify trajectories using Mutual Consistency.

Main Contributions

1. Rich 5-action MCTS search space.

2. Generator–Discriminator self-play reasoning.
3. State-of-the-art inference-time reasoning across 5 SLMs and 5 benchmarks.

Experimental Setup

- Tasks: Mathematical and commonsense reasoning.
- Datasets: GSM8K, GSM-Hard, MATH-500, SVAMP, StrategyQA.
- Baselines: CoT, Self-Consistency, ToT, RAP.
- Metric: Accuracy.

Experimental Results

- LLaMA2-7B: 12.5% → 63.9% on GSM8K.
- Mistral-7B: 81.9%.
- rStar consistently outperforms existing inference-time methods.

2. Background & Motivation

Problem Statement

- Although **LLMs** have achieved remarkable success, they still struggle with **complex multi-step reasoning**, especially on mathematical and logical tasks.
- This challenge is more severe for Small Language Models (SLMs), which often fail to effectively explore the solution space and cannot reliably evaluate or verify their own reasoning.
- Existing approaches often rely on **fine-tuning using stronger teacher models (e.g., GPT-4)** to improve reasoning.

Why important?

- Enables reasoning improvement without expensive fine-tuning or stronger teacher models.
- Makes **small language models more capable at inference time**, reducing computational and data requirements.
- Improves the practicality of deploying SLMs in resource-constrained environments

Limitations of Previous Work

- **Chain-of-Thought (CoT):** Explores only a single reasoning path, limiting exploration.
- **Self-Consistency (SC):** Requires many sampled reasoning paths, increasing inference cost.
- **RAP (Reasoning via Planning):** Uses only one reasoning action (sub-question generation), limiting search diversity.

- **Self-evaluation / Self-rewarding:** SLMs cannot reliably assess their own reasoning, leading to poor reward estimation and incorrect trajectory selection.
- **Fine-tuning methods:** Often depend on **strong teacher models** and additional training data.

3. Related Work

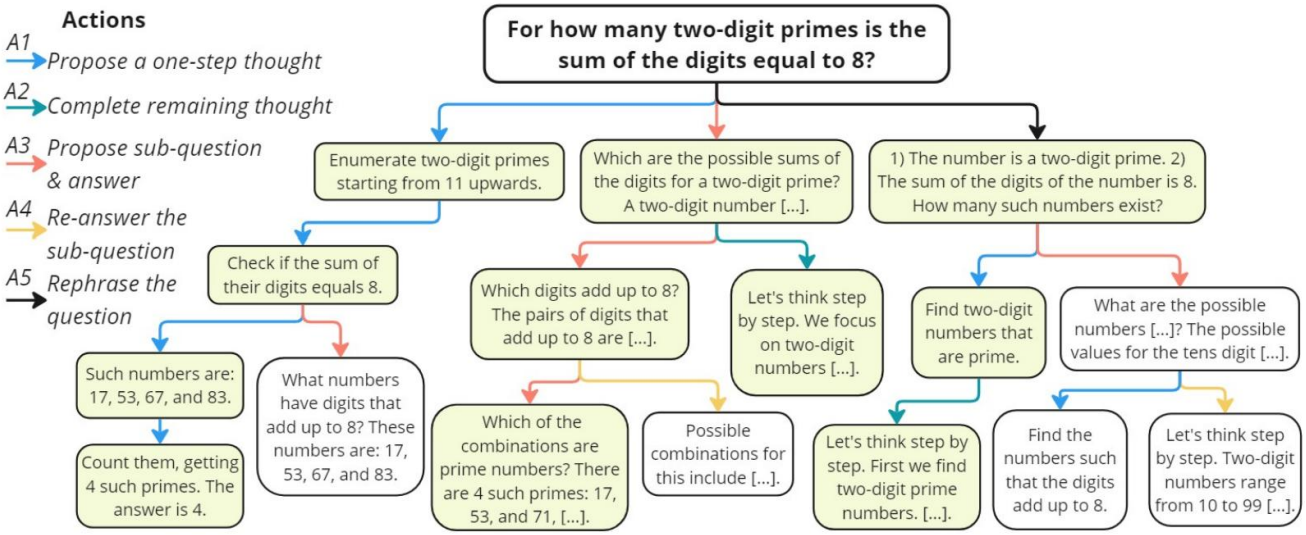
Existing Approaches

- Chain-of-Thought, Self-Consistency, Tree of Thoughts (ToT), RAP, reward models.

Differences

- rStar combines richer MCTS exploration with independent trajectory verification by another SLM.

4. Method



Overview

- The input question is initialized as the **root node** of the MCTS search tree.
- **MCTS selects a reasoning action** (A1–A5) according to its search policy.
- **SLM1 (Generator)** uses the selected action to generate the next reasoning step, expanding the search tree.
- After each rollout reaches a terminal node, the **reward is computed and back propagated** to update previous nodes.
- After multiple MCTS rollouts, a set of candidate reasoning trajectories is obtained.

- **SLM2 (Discriminator)** verifies each trajectory using **mutual consistency** based on partial reasoning hints.
- SLM1 selects the **highest-scoring validated trajectory** as the final answer.

Question: If a+b=8, b+c=-3, and a+c= -5, what is the value of the product abc?

Candidate Solution by SLM_1: Let's think step by step. From the three given equations, we can solve for a, b and c. Adding all three equations yields 2a + 2b + 2c = 0, or a + b + c = 0. Substituting a + c = -5 into this equation, we have -5 + b = 0, which implies b = 5. Similarly, substituting a + b = 8 gives a = 3, and using b + c = -3 results in c = -8. Finally, the product abc = 3 x 5 x (-8) = -120. The answer is: **-120**.

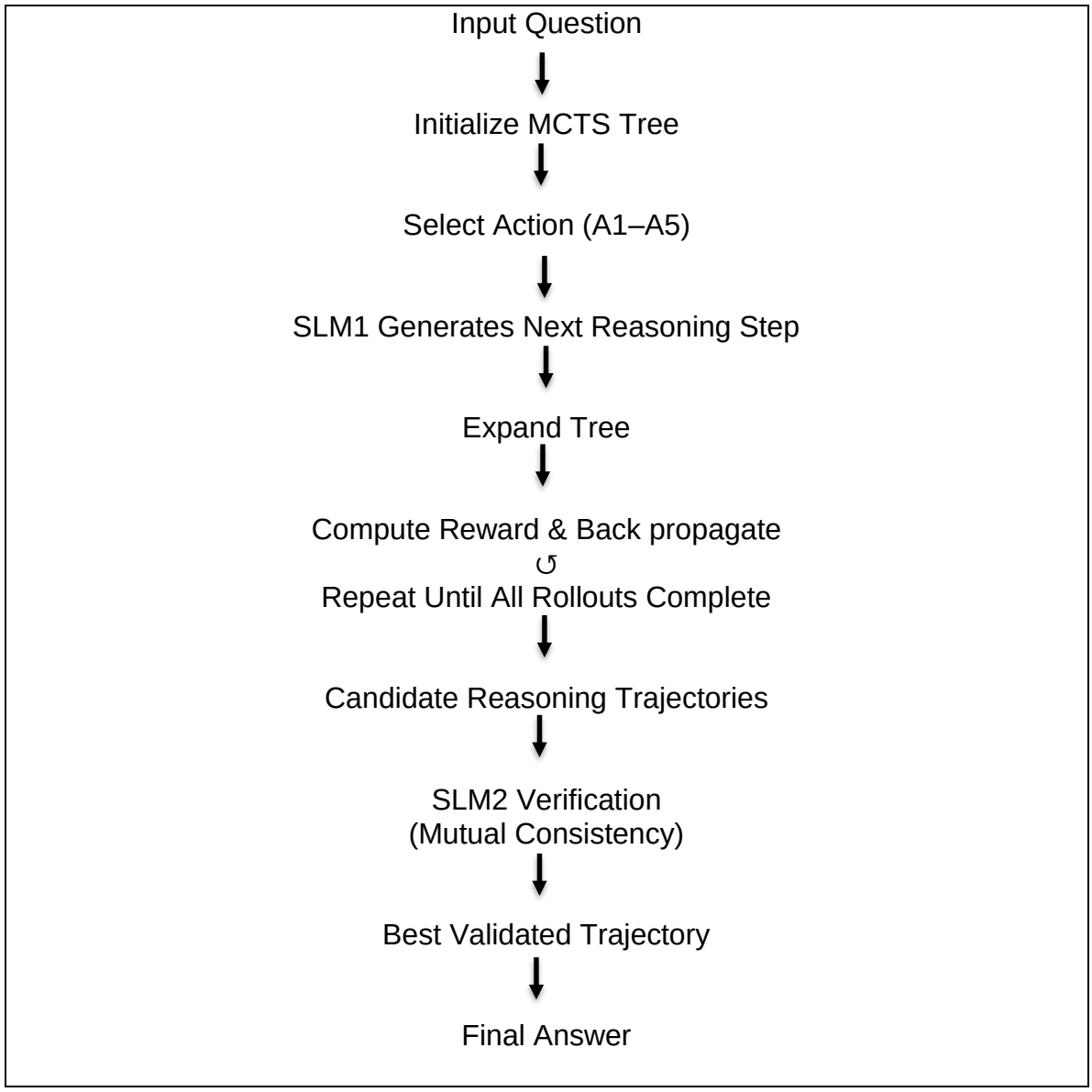
Masked Solution for SLM_2: Let's think step by step. From the three given equations, we can solve for a, b and c. Adding all three equations yields 2a + 2b + 2c = 0, or a + b + c = 0. Substituting a + c = -5 into this equation, we have -5 + b = 0, which implies b = 5. Similarly, substituting a + b = 8 gives a = 3, and using b + c = -3 results in c = -8. Finally, the product abc = 3 x 5 x (-8) = -120. The answer is: -120.

Completed Solution 1 by SLM_2: Let's think step by step. From the three given equations, we can solve for a, b and c. Adding all three equations yields 2a + 2b + 2c = 0, or a + b + c = 0. We can now use this result to find each variable individually. Using the equation a + b = 8, we substitute b + c = -3 to get a + (-3 - b) = 8 - 3, leading to b = 5. With b = 5, substituting into a + b = 8, we find a = 3. Then, using b = 5 in b + c = -3, we find c = -8. Finally, the product abc = 3 × 5 × (-8) = -120. The answer is: **-120**. **(consistent)**

Completed Solution 2 by SLM_2: Let's think step by step. From the three given equations, we can solve for a, b, and c. Adding all three equations yields 2a + 2b + 2c = 0, or a + b + c = 0. With a + c = -5, we have -5 + b = 0, so b = 5. Now, substituting b + c = -3, we deduce that c = -3 - 5 = -2. Finally, using a + b = 8, we find a = 8 - 5 = 3. The product abc = 3 × 5 × (-2) = -30. The answer is: **-30**. **(inconsistent)**

Figure 4: The prompt example for mutual reasoning consistency.

Complete Pipeline



Model Architecture

Generator SLM + MCTS + Discriminator SLM.

Key Components

MCTS	Explores multiple reasoning paths.
5 Reasoning Action(A1-A5)	Diversify reasoning strategies.
Reward	Guides MCTS search
Mutual Consistency	Verifies reasoning trajectories using a second SLM

Loss Functions

- None. No training or fine-tuning; inference only.

Training Strategy

- Inference-time only with 32 MCTS rollouts.

5. Experiments

Datasets

- GSM8K, GSM-Hard, MATH-500, SVAMP, StrategyQA.

Experimental Settings

- 5 SLMs, 32 rollouts, depth 5 (8 for MATH).

Evaluation Metrics

- Accuracy.

Baseline Methods

- Zero/Few-shot CoT, Self-Consistency, ToT, RAP.

6. Results & Analysis

Main Results

- Large improvements across all evaluated SLMs.

- Generator alone beats RAP and ToT; discriminator provides additional gains.

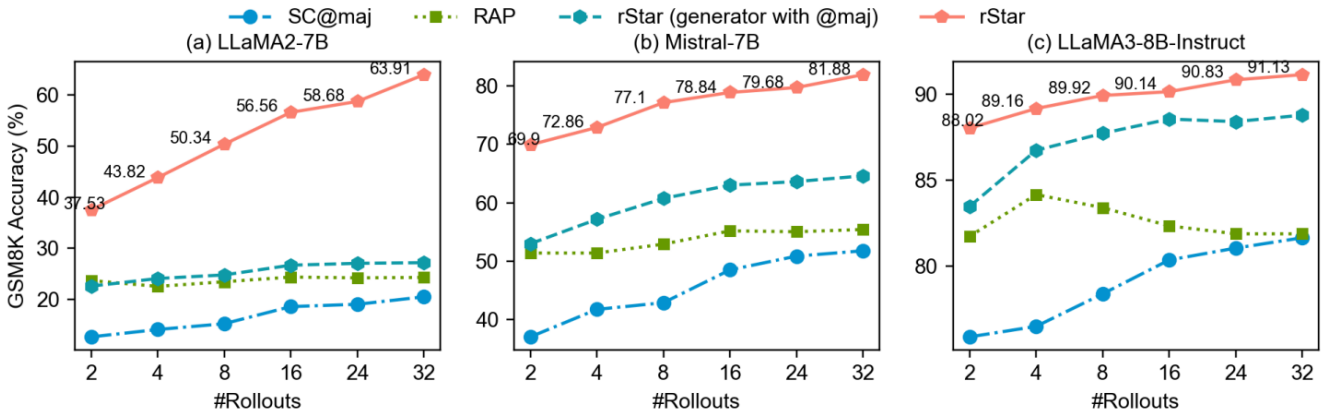


Figure 5: Performance comparison on the GSM8K dataset under different number of rollouts. rStar can significantly improve reasoning accuracy with just 2 rollouts.

Ablation Studies

- More rollouts improve performance.
- Five-action search outperforms single-action search.
- Mutual consistency outperforms self-verification.

Table 1: Ablation study on the effectiveness of our rich action space: we evaluate LLaMA3-8B on 200 sampled GSM8K questions.

Action Space	Accuracy
A_3 (i.e., RAP)	70.5
A_3 + A_5	72.5
A_3 +A_4 + A_5	73.5
A_2 + A_3+A_4 + A_5	74.0
All (A_1 + A_2 + A_3+A_4 + A_5)	**75.0**

Qualitative Analysis

- Better exploration and verification improve reasoning quality.

Failure Cases / Error Analysis

- Higher inference cost.

- Two SLMs can still agree on an incorrect answer.

7. Discussion

Strengths

- **No fine-tuning required:** Improves reasoning entirely at **inference time**, reducing training cost and avoiding dependence on stronger teacher models.
- **Novel reasoning framework:** Combines **MCTS-based exploration** with **mutual consistency** between two SLMs for better reasoning trajectory generation and verification.
- **Generalizable:** Demonstrated across **5 SLMs** and **5 diverse reasoning benchmarks**, showing consistent improvements.
- **Comprehensive evaluation:** Includes comparisons with strong baselines (CoT, Self-Consistency, ToT, RAP) and detailed ablation studies.

Limitations

- **High inference cost:** Multiple MCTS rollouts and an additional discriminator model increase inference latency and computational cost.
- **No correctness guarantee:** Two SLMs may still agree on an incorrect reasoning trajectory; mutual consistency does not guarantee correctness.
- **Limited evaluation domains:** Experiments focus mainly on mathematical and commonsense reasoning, leaving coding, planning, and scientific reasoning largely unexplored.

Future Work

- **Reduce inference cost** by developing more efficient search strategies or adaptive rollout policies.
- **Extend to additional domains**, such as code generation, scientific reasoning, planning, and multimodal tasks.

8. Personal Insights

Key Takeaways

- Reasoning quality can be improved without changing model weights.
- Search strategy is as important as model size.

Ideas Relevant to the Saegyeol AI Project

- The paper demonstrates that reasoning quality can be improved during inference without retraining. This idea could be integrated into SaeGyeol AI by adding a reasoning layer within the Private AI Control Plane, where private models explore multiple reasoning paths and verify them before producing the final response, improving both reliability and explainability.
- Search-based reasoning combined with independent verification would allow lightweight private models to achieve higher reasoning accuracy while reducing computational and training costs.