PRISM-MCTS: Learning from Reasoning Trajectories with Metacognitive Reflection

Siyuan Cheng, Bozhong Tian, YanChao Hao*, Zheng Wei*
Tencent PCG
{chancecheng,cobbtian,marshao,hemingwei}@tencent.com

Abstract

The emergence of reasoning models, exemplified by OpenAI-o1, signifies a transition from intuitive to deliberative cognition, effectively reorienting the scaling laws from pre-training paradigms toward test-time computation. Though Monte Carlo Tree Search (MCTS) is promising in this domain, existing methods are often inefficient, treating rollouts as isolated trajectories and causing significant computational redundancy. To address these limitations, we propose PRISM-MCTS, a novel reasoning framework that draws inspiration from human parallel thinking and reflective processes. PRISM-MCTS integrates a Process Reward Model (PRM) with a dynamic shared memory, capturing both *Heuristics* and *Fallacies*. By reinforcing successful strategies and pruning error-prone branches, PRISM-MCTS effectively achieves refinement. Furthermore, we develop a data-efficient training strategy for the PRM, achieving high-fidelity evaluation under a few-shot regime. Empirical evaluations across diverse reasoning benchmarks substantiate the efficacy of PRISM-MCTS. Notably, it halves the trajectory requirements on GPQA while surpassing MCTS-RAG and Search-o1, demonstrating that it scales inference by reasoning judiciously rather than exhaustively.

1 Introduction

In recent years, Large Language Models (LLMs) (Minaee et al., 2024; Zhao et al., 2023) have rapidly become a cornerstone in natural language processing, and the field of LLMs has been undergoing a paradigm shift. Conventional LLMs, such as GPT (OpenAI, 2023), LLaMA (Touvron et al., 2023), and Qwen (Yang et al., 2024), primarily rely on a "fast thinking" reasoning paradigm. This paradigm is primarily anchored in the knowledge and reasoning priors

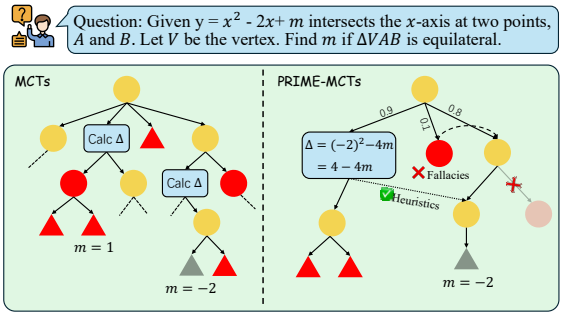

Figure 1: Standard MCTS (left) vs. PRISM-MCTS (right). PRISM-MCTS enables global information sharing to prune incorrect paths and reuse verified steps for higher efficiency.

synthesized during pre-training, facilitating rapid and intuitive generation from a single query. While "fast thinking" excels in routine tasks, it frequently falls short when addressing complex problems that necessitate multi-step iteration, rigorous logic, or dynamic adaptation to novel information. With the emergence of models like OpenAI-o1 (Jaech et al., 2024), "slow thinking" has become a pivotal direction for enhancing reasoning (Wei et al., 2022; Yao et al., 2023a; Li et al., 2025b), simulating human deliberation by scaling up inference-time compute.

Among the various pathways to achieving "slow thinking" reasoning techniques based on MCTS have become prominent (Hu et al., 2025; Qi et al., 2024; Zhang et al., 2024). By modeling reasoning as a structured tree search with backtracking, MCTS boosts complex logical task performance (Chen et al., 2024). Nevertheless, these approaches are encumbered by prohibitive computational overhead stemming from exhaustive exploration, while their reliance on a sequential paradigm induces informational isolation and computational waste through redundant node re-simulations. In resource-constrained or time-critical scenarios, balancing MCTS efficiency with

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 16392–16402

reasoning quality is the critical bottleneck for "slow thinking" techniques.

To address these limitations, we propose **P**rocess-**R**ewarded **I**ntelligent **S**hared **M**emory MCTS (PRISM-MCTS), as illustrated in Figure 1. Drawing from human metacognitive reflection, PRISM-MCTS captures the interconnected nature of complex reasoning trajectories, leveraging accumulated experience to distill valuable guidance from even erroneous paths. Consequently, these distilled insights effectively optimize the navigation of both concurrent and subsequent search trajectories. Specifically, PRISM-MCTS utilizes a PRM to conduct granular assessments of each reasoning step, subsequently isolating nodes with divergent rewards into a dynamic memory for classification as heuristics or fallacies. During concurrent reasoning, knowledge from memory nodes is shared globally to foster system-wide synergy. Specifically, *Heuristics Memory* empowers subsequent nodes to exploit verified logic from antecedent trajectories to accelerate convergence, whereas *Fallacies Memory* serves as a proactive constraints that prevent the redundant expenditure of computational resources on analogous errors. Moreover, we employ a Memory Manager to efficiently distill key information from the repository's state. Through these mechanisms, PRISM-MCTS shifts the traditional serial reasoning paradigm toward a parallelized search framework supported by global information sharing. Grounded in reflection and experiential sharing, this mechanism refines MCTS selectivity, thereby bolstering both reasoning efficiency and deductive rigor in intricate tasks.

In summary, the main contributions of our work are as follows:

- We introduce PRISM-MCTS, a parallelized reasoning framework that leverages a reflective memory mechanism to share global heuristics and prune fallacies, thereby optimizing search efficiency.
- We devise a dual-stage, few-shot PRM training method synergizing Step-wise DPO and multi-class classification for precise, granular process evaluation.
- Extensive experiments demonstrate that PRISM-MCTS outperforms state-of-the-art baselines on complex reasoning tasks, achieving higher accuracy with significantly fewer search steps.

2 Related Work

Inference-time Scaling. Scaling inference-time computation has emerged as a compelling paradigm for enhancing the reasoning capabilities of LLMs (OpenAI, 2024; Guo et al., 2025; Hao et al., 2023; Lightman et al., 2023). Early efforts like Chain-of-Thought (CoT) prompting (Wei et al., 2022) focused on linear trajectories, encouraging models to generate intermediate reasoning steps. To improve the robustness of linear reasoning, Self-Consistency (CoT-SC) (Wang et al., 2023) samples diverse reasoning paths and selects the final answer via majority voting. Additionally, Program-of-Thought (PoT) (Chen et al., 2023) generates executable code to offload computational sub-tasks to external interpreters. However, these linear methods suffer from error propagation and limited flexibility. To overcome these limitations, tree-based planning methods like Tree-of-Thought (ToT) (Yao et al., 2023a), Graph-of-Thought (GoT) (Besta et al., 2024), and Monte Carlo Tree Search (MCTS) enable structured search with backtracking and evaluation, significantly improving complex reasoning (Zhou et al., 2023; Gan et al., 2025; Qi et al., 2024).

MCTS with Process Reward Models. MCTS provides a principled framework for balancing exploration and exploitation, demonstrating remarkable efficacy in domains requiring rigorous logic, such as mathematics and code generation (Zhou et al., 2024; Zhang et al., 2023; Chen et al., 2024; Xu et al., 2025). However, terminal outcome supervision causes inefficient exploration, as MCTS lacks intermediate guidance in vast search spaces. To address this, recent studies have integrated PRMs directly into the MCTS framework to provide dense, step-level guidance. By leveraging granular feedback to reduce error propagation, methods such as ReST-MCTS* (Zhang et al., 2024), AR-MCTS (Dong et al., 2025), and I-MCTS (Liang et al., 2025) significantly enhance the effectiveness of the search process. However, training effective PRM typically necessitates extensive supervision, posing significant challenges for data-scarce domains (Lightman et al., 2024; Uesato et al., 2022; Zhang et al., 2025). In this work, we propose a dual-stage, data-efficient training strategy tailored for MCTS-guided reasoning.

By combining step-wise preference learning with fine-grained value classification, PRISM-MCTS enables precise few-shot evaluation, enhancing guided search scalability and effectiveness.

Reflective and Memory-Augmented Reasoning. Integrating memory mechanisms enables LLMs to transcend the limitations of static parametric knowledge, facilitating context retention and iterative refinement (Shinn et al., 2023; Packer et al., 2023; Fang et al., 2025). While early approaches like Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) and IRCoT (Trivedi et al., 2023) primarily focused on enhancing factual accuracy via external corpora, they often function as static lookups lacking dynamic adaptation during reasoning. To enhance structured reasoning, recent studies have begun to incorporate these reflective and memory capabilities directly into MCTS frameworks. For instance, MC-DML (Shi et al., 2025) and I-MCTS (Liang et al., 2025) employ introspection to refine intermediate steps, while CoAT (Pan et al., 2025) leverages associative memory to link related concepts during inference. However, these methods typically aggregate experiences indiscriminately, failing to explicitly distinguish between valid logical patterns and distinct failure modes, which can lead to error recurrence. In this work, we propose a structured dual-memory mechanism*Heuristics Memory* for reinforcing successful strategies and *Fallacies Memory* for intercepting erroneous paths. Governed by a *Memory Manager*, this architecture dynamically filters high-fidelity insights to proactively prune redundant branches, transforming the search into a self-improving reasoning system.

3 PRISM-MCTS

3.1 Preliminaries

Monte Carlo Tree Search. MCTS is a widely used sampling-based search method for decision-making optimization. It constructs a search tree by repeatedly executing four steps: selection, expansion, simulation, and back-propagation. During the selection phase, MCTS recursively selects child nodes from the root using the Upper Confidence Bounds applied to Trees (UCT) (Kocsis and Szepesvári, 2006). The UCT of a node n is calculated as follows:

$$UCT(n) = V(n) + \epsilon\sqrt{\frac{lnN(p)}{N(n)}}. \quad (1)$$

Algorithm 1 PRISM-MCTS Search Process.

Require: Root node S_{root}, number of rollouts $\mathcal{M}$, exploration constant ϵ, memory thresholds τ_{pos}, τ_{low}
Ensure: Updated search tree statistics V, N, memory states $\mathcal{MEM}_H, \mathcal{MEM}_F$

Initialize $\mathcal{MEM}_H \leftarrow \emptyset, \mathcal{MEM}_F \leftarrow \emptyset$
for $j = 1$ to $\mathcal{M}$ **do**
  $C \leftarrow S_{root}$
  ——————-Selection——————-
  while C is not leaf node **do**
    $C \leftarrow argmax_{C' \in children(C)} \left(\mathcal{V}_{C'} + \epsilon\sqrt{\frac{2\ln N_C}{N_{C'}}}\right)$
  end while
  ——————-Expansion——————-
  if C is not terminal **then**
    $C^* \leftarrow FindChilds(C, \mathcal{MEM}_F, \mathcal{MEM}_H)$
    for each child c in C^* **do**
      $Update(\mathcal{V}_c)$
    end for
  end if
  ——————-Simulation——————-
  while C is not terminal **do**
    if C is not full Expanded **then**
      $Expand(C)$
    end if
    for each c in $children(C)$ **do**
      if $\mathcal{V}_c \geq \tau_{pos}$ **then** Add c to $\mathcal{MEM}_H$
      if $\mathcal{V}_c \leq \tau_{neg}$ **then** Add c to $\mathcal{MEM}_F$
    end for
    $C \leftarrow argmax_{C' \in children(C)}(\mathcal{V}_{C'})$
  end while
  ——————Backpropagation——————
  $UpdateStatistics(N_{C'})$ ▷ Update all path nodes
end for
  return Search Tree

where $N(n)$ is the number of visits to node n, $V(n)$ is the score value, and p is the parent node of node n. ϵ is the exploration weight. Upon episode completion, a back-propagation is performed to update the value of node n and its parent nodes.

Problem Formulation. We define the task as generating a sequence of steps to solve a given problem x. The solution is represented as a trajectory $p_T = [s_1, s_2, \ldots, s_T]$, where each s_t denotes a reasoning step and s_T includes the final answer. The objective is to identify the optimal trajectory that maximizes the likelihood of correctness.

3.2 Human-Like Reflective Memory

PRISM-MCTS integrates a memory mechanism inspired by human reflection. As shown in Figure 2, this system comprises three essential components: **Heuristics Memory**, **Fallacies Memory**, and a **Memory Manager**. The overall search procedure is summarized in Algorithm 1.

Heuristics Memory. The Heuristics module ($\mathcal{MEM}_H$) archives optimal reasoning trajectories and validated intermediate states. During MCTS

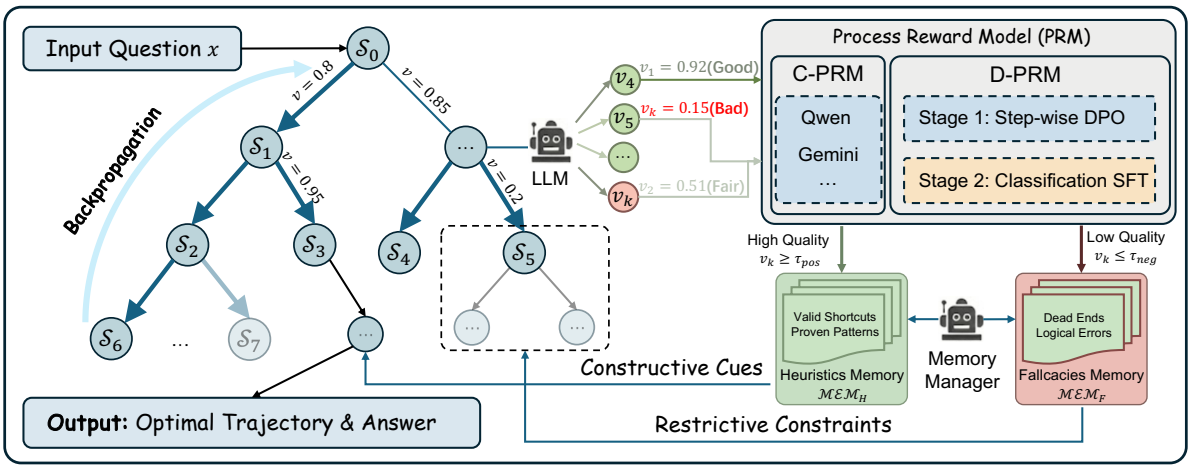

Figure 2: Overview of the PRISM-MCTS framework. It integrates MCTS with a PRM (Continuous-PRM or Discrete-PRM) and a reflective memory system. Stored **Heuristics Memory** and **Fallacies Memory** facilitate global information sharing to guide generation and prune the search space efficiently.

simulation, if a node's evaluated value v_k exceeds a predefined positive threshold τ_{pos}, its content is integrated into $\mathcal{MEM}_H$. These stored insights serve as guiding signals for subsequent expansions to facilitate the replication of successful strategies.

Fallacies Memory. The Fallacies module ($\mathcal{MEM}_F$) serves as a repository for erroneous reasoning patterns and failed trajectories. When a nodes value v_k falls below a negative threshold τ_{neg}, it is archived within $\mathcal{MEM}_F$ to provide negative feedback. This mechanism provides essential constraints for pruning the search space, enabling the model to circumvent recurrent logical errors and improve search throughput.

Memory Manager. The Memory Manager orchestrates interactions between search and memory modules to maintain informational parsimony. Its primary function is to distill pivotal patterns while filtering out redundant entries, thereby preventing computational overhead. By ensuring that only high-fidelity insights remain in $\mathcal{MEM}_H$ and $\mathcal{MEM}_F$, the manager optimizes the quality of guidance provided during the reasoning process.

3.3 Dual-Stage Process Reward Model

Inspired by AR-MCTS (Dong et al., 2025), we adopt a two-stage training strategy for our PRM. However, unlike previous approaches that rely on large-scale datasets, our method is tailored for **few-shot** scenarios where data scarcity is a challenge. To address this, we introduce a classification-based objective in the second stage.

Data Construction. We adopt the self-training pipeline from ReST-MCTS* (Zhang et al., 2024) to construct high-quality data. Based on the search tree generated by MCTS, we calculate a quality value v_k for each partial solution $p_k = [s_1, \ldots, s_k]$, which serves as the ground truth. The value v_k is iteratively updated to reflect the cumulative progress towards the correct answer:

$$v_k = \max(v_{k-1} + w_{s_k}, 0) \tag{2}$$

where w_{s_k} is the weighted reward for step s_k. To ensure the value reflects both step correctness and reasoning progress, w_{s_k} incorporates the reasoning distance m_k (minimum steps to the correct answer) and a step-level error indicator r_{s_k}:

$$w_{s_k} = \frac{1 - v_{k-1}}{m_k + 1}(1 - 2r_{s_k}) \tag{3}$$

Here, r_{s_k} is derived from the search tree (set to 0 if the step is on a correct path, 1 otherwise).

Preference Alignment via SDPO. In the first stage, we align the model's preferences using Step-level Direct Preference Optimization (SDPO). By leveraging the reasoning paths generated during MCTS, we construct preference pairs (y^+, y^-), where y^+ represents a high-quality step ($v_k \geq 0.8$) and y^- represents a low-quality or incorrect step ($v_k \leq 0.2$). The training objective is to maximize the likelihood of the preferred step over the dispre-

ferred one relative to a reference model π_{ref}:

$$\mathcal{L}_{\text{SDPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(s_i, y^+, y^-)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y^+|s_i)}{\pi_{\text{ref}}(y^+|s_i)} - \beta\log\frac{\pi_\theta(y^-|s_i)}{\pi_{\text{ref}}(y^-|s_i)}\Big)\Big] \tag{4}$$

where β is a hyperparameter controlling the deviation from the reference model.

Fine-Grained Value Classification. In the second stage, we reframe value estimation as classification rather than regression. This shift is motivated by the intuition that discrete categories generalize better than continuous values in few-shot settings. Specifically, we discretize the original scores $v_k \in [0, 1]$ into five categories $\mathcal{C} = \{\text{Perfect, Good, Fair, Poor, Bad}\}$ using a uniform interval of 0.2. Given a step s_i and its ground truth label $y_i \in \mathcal{C}$, we optimize the cross-entropy loss:

$$\mathcal{L}_{\text{CLS}} = -\sum_{c\in\mathcal{C}} \mathbb{I}(y_i = c)\log P_\theta(c|s_i) \tag{5}$$

where $P_\theta(c|s_i)$ is the probability assigned to class c by the PRM.

4 Experiments

4.1 Benchmarks and Metrics

To evaluate the effectiveness of PRISM-MCTS, we selected two types of benchmarks designed to stress-test the models performance across both knowledge depth and logical complexity:

Science and Fact Verification. This category assesses knowledge-intensive processing and information discernment via: (1) GPQA Diamond (Rein et al., 2023), which focuses on high-level scientific question answering; and (2) FoolMeTwice (FMT) (Eisenschlos et al., 2021), a challenging fact-checking benchmark that requires the verification of complex factual claims.

Mathematical Reasoning. This category evaluates the model's capacity for rigorous logical deduction via: (1) MATH500 (Lightman et al., 2024), where we specifically select the most challenging problems (Level 5) to test multi-step problem-solving across competition-level mathematics; and (2) AIME25 (Zhang and Math-AI, 2025), which utilizes problems from the American Invitational Mathematics Examination to challenge the model's advanced reasoning limits.

Evaluation Metrics. For Science and Fact Verification, we adopt two widely used metrics: Exact Match (EM), and F1 scores. For Mathematical Reasoning, performance is measured by Accuracy (Score), with MCTS-based methods deriving the final answer by selecting the maximum probability. Furthermore, to validate the efficiency of MCTs, we introduce two structural metrics: **Trajectory:** Quantifies the expansion breadth of the search tree to assess exploration range. **Depth:** Measures the expansion layers to evaluate the length and complexity of the reasoning chains.

4.2 Baseline Systems

To assess the efficacy of PRISM-MCTS, we benchmark it against a set of frontier reasoning baselines, spanning from prompting techniques to sophisticated agentic and search-based frameworks.

Zero-Shot CoT. We employ standard Zero-Shot CoT prompting to elicit reasoning capabilities from the LLMs without task-specific examples.

ReAct. ReAct (Yao et al., 2023b) synergizes reasoning and acting by interleaving the generation of reasoning traces and task-specific actions. This allows the model to dynamically retrieve external information to support its reasoning process, enabling it to refine its understanding based on external evidence.

Search-o1. Search-o1 (Li et al., 2025a) enhances large reasoning models with an agentic RAG mechanism. It integrates a "Reason-in-Documents" module that refines retrieved documents into concise reasoning steps, allowing the model to autonomously retrieve and integrate external knowledge on demand while maintaining reasoning coherence.

Rest-MCTs. Rest-MCTs (Zhang et al., 2024) introduces a process reward model (PRM) guided tree search policy. It utilizes a self-training pipeline where the policy model and PRM are iteratively improved using high-quality traces generated via MCTS. This method focuses on enhancing internal reasoning capabilities through search and self-improvement.

MCTS-RAG. MCTS-RAG (Hu et al., 2025) integrates Monte Carlo Tree Search with Retrieval-Augmented Generation. It expands the action space of standard MCTS to include retrieval-specific actions, enabling the model to explore

Model	Method	GPQA-DIAMOND		FMT		MATH500	AIME25
		EM ↑	F1 ↑	EM ↑	F1 ↑	Score ↑	Score ↑
GPT-4.1-mini	Zero-shot	47.55	46.55	68.34	68.75	79.25	48.67
	ReAct	55.05	60.75	65.00	67.61	79.10	50.00
	Search-O1	57.07	57.16	66.00	65.03	80.59	43.33
	ReST-MCTS*	60.61	60.53	69.00	68.91	70.90	20.00
	MCTS-RAG	64.65	64.80	69.50	68.98	80.60	50.00
	PRISM-MCTS	**65.08**	**65.16**	**70.50**	**70.07**	**82.09**	**53.33**
Qwen3-30B-A3B-2507	Zero-shot	45.80	45.29	62.72	62.85	75.07	53.33
	ReAct	48.99	53.49	65.50	65.43	70.90	50.00
	Search-O1	61.11	53.76	68.00	67.76	88.06	50.00
	ReST-MCTS*	58.08	60.45	64.00	63.43	77.61	20.00
	MCTS-RAG	63.64	64.26	**70.50**	**69.99**	**94.03**	**66.67**
	PRISM-MCTS	**65.15**	**66.26**	68.00	67.41	93.28	63.33

Table 1: Main performance comparison of PRISM-MCTS against various baselines on scientific verification (GPQA-Diamond, FMT) and mathematical reasoning (MATH, AIME25) benchmarks. Best results for each backbone model are highlighted in **bold**.

multiple reasoning and retrieval trajectories. This allows for dynamic knowledge acquisition and the selection of the most consistent reasoning path through voting mechanisms.

To ensure fair comparisons, we use the same base LLMs: Qwen3-30B-A3B-Instruct-2507 and GPT-4.1-mini for all evaluated systems. We keep all hyperparameters consistent (e.g., ϵ, τ_{pos}, τ_{neg}).

4.3 Main Results

Performance on Science and Fact Verification. As shown in Table 1. In knowledge-intensive tasks, PRISM-MCTS exhibits remarkable robustness. On the challenging GPQA-Diamond benchmark, PRISM-MCTS achieves state-of-the-art results. With GPT-4.1-mini, PRISM-MCTS reaches an EM score of 65.08%, significantly outperforming the standard Zero-shot baseline and the strong retrieval-augmented baseline MCTS-RAG. Similarly, using the Qwen3-30B-A3B-Instruct-2507 backbone, PRISM-MCTS achieves the highest EM of 65.15%, surpassing Search-O1 and MCTS-RAG. On the FMT, PRISM-MCTS with GPT-4.1-mini achieves the best performance with an EM of 70.50% and F1 of 70.07%, effectively mitigating hallucination risks common in direct reasoning. While MCTS-RAG shows slightly higher scores on FMT with Qwen3-30B-A3B-Instruct-2507, PRISM-MCTS remains highly competitive and significantly outperforms Zero-shot and ReAct baselines. However, the primary advantage of PRISM-MCTS lies in its search efficiency(§4.4).

Performance on Mathematical Reasoning. In the domain of logical deduction, PRISM-MCTS demonstrates strong reasoning capabilities. On MATH500 and AIME25, PRISM-MCTS with GPT-4.1-mini secures the top position, achieving scores of 82.09 and 53.33, respectively, surpassing both ReAct and MCTS-RAG. Notably, on the AIME25 benchmark, which tests advanced reasoning limits, PRISM-MCTS shows a clear advantage over Search-O1. For the Qwen3-30B-A3B-Instruct-2507, PRISM-MCTS maintains high performance, comparable to the best-performing baseline MCTS-RAG and significantly exceeding Zero-shot and Search-O1. This indicates that integrating MCTS with our proposed mechanism enhances the model's ability to navigate complex solution spaces in mathematical problems.

Across all tasks, PRISM-MCTS shows consistent improvements over the Zero-shot and ReAct baselines, validating the efficacy of integrating MCTS-based planning with retrieval augmentation. The substantial gains in GPQA and the competitive results in math benchmarks highlight the versatility of PRISM-MCTS in handling both knowledge-retrieval heavy and logic-heavy tasks.

4.4 Search Efficiency Analysis

To further assess the computational efficiency of PRISM-MCTS, we analyze the structural characteristics of the generated search trees compared to MCTS-RAG. Specifically, we focus on two key metrics: **Trajectories**, which measures the breadth of exploration (i.e., the number of valid reasoning paths explored), and **Depth**, which indicates the average length of the reasoning chains. Figure 3 illustrates these statistics across four benchmarks using both GPT-4.1-mini and Qwen3-

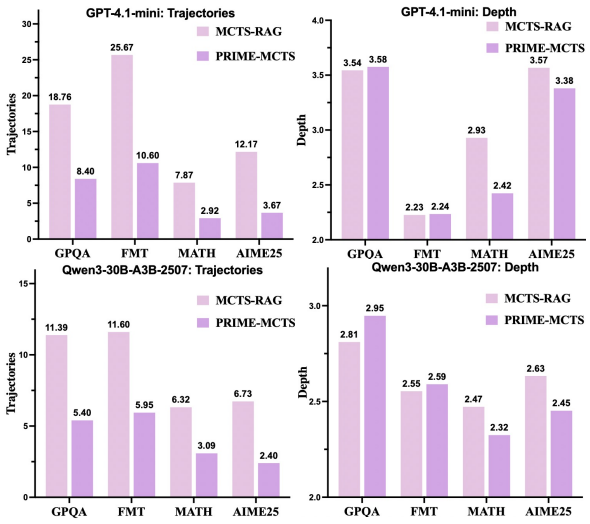

Figure 3: Comparison of search efficiency between MCTS-RAG and PRISM-MCTS. The left panels display the average number of explored Trajectories (search breadth), while the right panels show the average reasoning Depth. PRISM-MCTS consistently reduces the search space while maintaining optimized reasoning paths.

30B-A3B-Instruct-2507 as backbones.

Reduced Search Trajectories. As shown in the left panels of Figure 3, PRISM-MCTS demonstrates a significant reduction in the number of trajectories compared to MCTS-RAG across all datasets. For instance, on the GPQA benchmark with GPT-4.1-mini, PRISM-MCTS reduces the average number of trajectories from 18.76 to 8.40, a decrease of over 55%. Similarly, on the AIME25 dataset with Qwen3-30B-A3B-Instruct-2507, the trajectories drop from 6.73 to 2.40. This substantial reduction indicates that PRISM-MCTS effectively prunes the search space. By leveraging the Memory mechanism through Heuristics ($\mathcal{MEM}_H$) and Fallacies ($\mathcal{MEM}_F$), the model identifies and prunes suboptimal or redundant trajectories during the search process. This allows PRISM-MCTS to concentrate its computational resources on the most plausible reasoning directions instead of brute-force tree expansion.

Optimized Reasoning Depth. Regarding reasoning depth, PRISM-MCTS maintains a comparable or slightly optimized depth relative to MCTS-RAG. On knowledge-intensive tasks like GPQA and FMT, the depth remains stable, ensuring that the reasoning rigor is preserved. In mathematical tasks like MATH500 and AIME25, we observe a slight reduction in depth, suggesting that the memory-augmented guidance helps the model find more direct and efficient solutions without unnecessary intermediate steps.

	EM↑	F1↑	LA↑	Traj↓	Depth↓
GPT-4.1-mini					
Oracle-PRM	70.50	**70.07**	76.00	**10.60**	**2.24**
Light-PRM	**70.50**	69.91	**80.50**	14.91	2.33
Qwen3-30B-A3B-2507					
Oracle-PRM	68.00	67.41	**82.50**	**5.95**	2.59
Light-PRM	**68.00**	**67.41**	79.50	6.65	**2.52**

Table 2: Comparison of reasoning performance and search efficiency between the frontier Gemini-2.5-Pro and our proposed locally-trained dual-stage PRM.

Conclusion. The combination of significantly reduced search breadth and stable reasoning depth confirms the efficiency of PRISM-MCTS. It achieves superior performance (as shown in Table 1) while consuming fewer search steps, demonstrating that the integration of parametric memory into the MCTS framework guides the model to reason "smarter" rather than just "harder."

4.5 Dual-Stage Process Reward Model

To evaluate the effectiveness of our two-stage training strategy (SDPO and Fine-Grained Classification), we investigate whether the distilled local Dual-Stage Process Reward Model can match the performance of high-capability closed-source models. Specifically, we compare two configurations of the reward evaluator: (1) **Oracle-PRM**: Utilizing Gemini-2.5-Pro (Team, 2025) to provide reward scoring and memory management, representing a high-performance upper bound. (2) **Light-PRM**: Employing our locally fine-tuned Qwen-3-4B model, trained via the two-stage pipeline, to perform these auxiliary roles. Table 2 presents the comparative results using both GPT-4.1-mini and Qwen3-30B-A3B-Instruct-2507 as the core reasoning backbones.

Reward Capability Alignment. Remarkably, the **Light-PRM** achieves performance parity with the **Oracle**-guided system. With the GPT-4.1-mini backbone, the distilled model matches the oracle exactly in Exact Match (EM) at 70.50% and maintains a highly competitive F1 score (69.91 vs. 70.07). Similarly, under the Qwen3-30B-A3B-Instruct-2507 backbone, the local model achieves identical EM (68.00%) and F1 (67.41%). These results demonstrate that our proposed self-training

pipeline effectively distills the critical discrimination and planning capabilities, such as identifying valid reasoning steps and potential pitfalls.

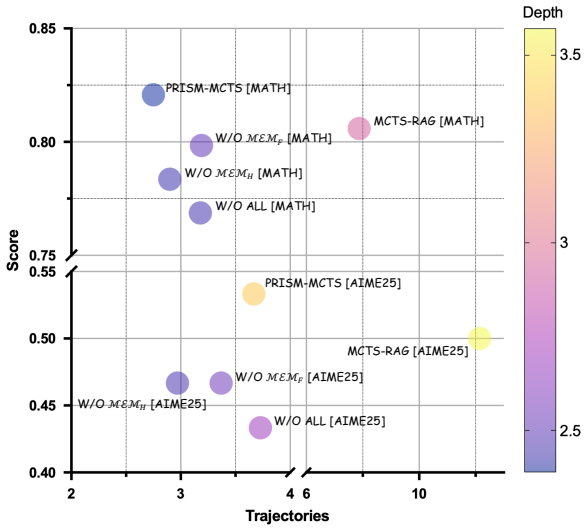

Figure 4: Ablation study on MATH500 and AIME25 benchmarks evaluating the impact of Heuristics Memory ($\mathcal{MEM}_H$) and Fallacies Memory ($\mathcal{MEM}_F$) modules. The scatter plot correlates search efficiency (Trajectories) with reasoning performance (Score), where color indicates reasoning Depth. Removing memory components leads to performance degradation, while the full PRISM-MCTS maintains high accuracy with minimal search breadth, significantly outperforming the MCTS-RAG baseline in efficiency.

Efficiency Analysis. In terms of search efficiency, the Local model exhibits a slightly larger search breadth compared to the Gemini oracle. As shown in the *Trajectories* column of Table 2, the average number of trajectories for the Local model is higher. This suggests that the local PRM is slightly less discriminative in the early stages of search, necessitating a broader exploration to locate the optimal solution. However, the reasoning *Depth* remains stable, confirming that the local guidance successfully prevents the reasoning from degenerating into inefficient, overly long chains. The minimal performance gap between the Local and Gemini configurations highlights the robustness of PRISM-MCTS. It offers a flexible trade-off: users can leverage powerful APIs for maximum efficiency, or opt for the local deployment to achieve efficient reasoning capabilities with full data privacy and reduced operational costs.

4.6 Ablation Studies

We perform ablation studies on MATH500 and AIME25 benchmarks to evaluate the specific contributions of $\mathcal{MEM}_H$ and $\mathcal{MEM}_F$. We compare the full PRISM-MCTS model against three configurations: (1) w/o $\mathcal{MEM}_H$, where the Heuristics module is disabled; (2) w/o $\mathcal{MEM}_H$, where the Fallacies module is removed; and (3) w/o ALL, where both memory components are deactivated.

Figure 4 illustrates the results, where the x-axis represents the search breadth (Trajectories), the y-axis represents the model performance (Score), and the marker color corresponds to the reasoning Depth (as shown in the color bar).

Impact of Memory on Efficiency and Accuracy. The most striking observation is the clustering of PRISM-MCTS and its ablation variants on the left side of the plot, maintaining high performance with very few trajectories (≈ 3). In contrast, the MCTS-RAG baseline relies on a significantly broader search, requiring roughly $2\times$ to $4\times$ more trajectories to achieve comparable or even lower scores. In contrast, PRISM-MCTS achieves higher (or comparable) scores with substantially fewer trajectories. However, when memory is fully removed, the performance drops drastically (e.g., on MATH500, from $\sim$0.82 to $\sim$0.77). This finding demonstrates that the efficiency of PRISM-MCTS is not a mere byproduct of breadth restriction; instead, the Memory mechanism provides the critical navigational guidance required to pinpoint correct solutions within a compact search space.

Relative Importance of Heuristics and Fallacies. We observe that both Heuristics Memory and Fallacies Memory are essential for peak performance, though their relative impact varies. On the MATH500 benchmark, deactivating Heuristics (w/o $\mathcal{MEM}_H$) causes a more pronounced decline than removing Fallacies (w/o $\mathcal{MEM}_F$). This suggests that positive reinforcement through successful trajectories is marginally more critical for mathematical reasoning than error avoidance. Ultimately, the synergy between these modules yields the most robust reasoning policy by balancing constructive guidance with negative constraints.

5 Conclusion

In this paper, we present PRISM-MCTS, a framework that advances inference-time scaling by integrating a PRM with a reflective shared memory system. By dynamically leveraging *Heuristics Memory* and *Fallacies Memory*, PRISM-MCTS delivers strong performance on complex bench-

marks while substantially reducing the search breadth. Furthermore, we propose a resource-efficient PRM training strategy enabling local models to rival the fidelity of top closed-source PRM model. Overall, our method demonstrates that through memory reflection, we can effectively enhance MCTS search efficiency and achieve higher-quality reasoning.

Limitations

Limited PRM Training Scale. Due to resource constraints, the volume of synthesized training data for the Process Reward Model was restricted. We hypothesize that if sufficient computational resources were available to generate a larger corpus of high-quality process supervision data, the model would demonstrate superior domain adaptation capabilities, thereby leading to more robust reasoning performance.

Absence of Multi-modal Support. Our current implementation and evaluation of PRISM-MCTS are primarily confined to text-based reasoning tasks. While the proposed reflective memory mechanism and PRM-guided search framework are theoretically applicable to multi-modal contexts (e.g., visual reasoning or diagram-based mathematical problem solving), the current system has not yet been extended to handle non-textual inputs. As advanced reasoning tasks increasingly require the integration of cross-modal information, the lack of multi-modal capabilities remains a limitation of the current work.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: Process supervision without process. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Trans. Mach. Learn. Res.*, 2023.

Guanting Dong, Chenghao Zhang, Mengjie Deng, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. 2025. Progressive multimodal reasoning via active retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3579–3602. Association for Computational Linguistics.

Julian Martin Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan L. Boyd-Graber. 2021. Fool me twice: Entailment from wikipedia gamification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 352–365. Association for Computational Linguistics.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. 2025. Lightmem: Lightweight and efficient memory-augmented generation. *CoRR*, abs/2510.18866.

Bingzheng Gan, Yufan Zhao, Tianyi Zhang, Jing Huang, Yusu Li, Shu Xian Teo, Changwang Zhang, and Wei Shi. 2025. MASTER: A multi-agent system with LLM specialized MCTS. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 9409–9426. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruite Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Z. Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua J Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Yunhai Hu, Yilun Zhao, Chen Zhao, and Arman Cohan. 2025. MCTS-RAG: enhancing retrieval-augmented generation with monte carlo tree search. *CoRR*, abs/2503.20757.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. Openai o1 system card. *CoRR*, abs/2412.16720.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K"uttler, Mike Lewis, Wen-tau Yih, Tim Rockt"aschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2025b. From system 1 to system 2: A survey of reasoning large language models. *CoRR*, abs/2502.17419.

Zujie Liang, Feng Wei, Wujiang Xu, Lin Chen, Yuxi Qian, and Xinhui Wu. 2025. I-MCTS: enhancing agentic automl via introspective monte carlo tree search. *CoRR*, abs/2502.14693.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Edwards Harrison, Armond Rivers, Bowen Baker, M. Balaji, Maciej Besta, Greg Brockman, Brooke Chen, and 1 others. 2023. Let's reward step-by-step: Step-level reward model as a verifier for mathematical reasoning. *arXiv preprint arXiv:2305.20050*.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *CoRR*, abs/2402.06196.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024. Learning to reason with llms. *https://openai.com/index/learning-to-reason-with-llms/*.

Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Ion Stoica. 2023. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Jianfeng Pan, Senyou Deng, and Shaomang Huang. 2025. Coat: Chain-of-associated-thoughts framework for enhancing large language models reasoning. *CoRR*, abs/2502.02390.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *CoRR*, abs/2408.06195.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022.

Zijing Shi, Meng Fang, and Ling Chen. 2025. Monte carlo planning with large language model for text-based game agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Noah Shinn, Federico Labash, and Ashwin Gopinath. 2023. Reflexion: Language agents with iterative self-reflection and episodic memory. *arXiv preprint arXiv:2303.11366*.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Afroz Maarten, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Bin Xu, Yiguan Lin, Yinghao Li, and Yang Gao. 2025. SRA-MCTS: self-driven reasoning augmentation with monte carlo tree search for code generation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 8678–8686. ijcai.org.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Dan Zhang, Sining Zhou, Ziyang Wu, Shibo Hao, and Zhiting Hu. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023. Planning with large language models for code generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 10495–10516. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Andy Zhou, Kai Yan, Michail Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search with reasoning and action. *arXiv preprint arXiv:2310.04406*.

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 *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.