VReST: Enhancing Reasoning in Large Vision-Language Models through Tree Search and Self-Reward Mechanism

Anonymous ACL submission

Abstract

Large Vision-Language Models (LVLMs) have shown exceptional performance in multimodal tasks, but their effectiveness in complex visual reasoning is still constrained, especially when employing Chain-of-Thought prompting techniques. In this paper, we propose **VReST**, a novel training-free approach that enhances **Re**asoning in **L**VLMs through Monte Carlo **T**ree Search and **S**elf-Reward mechanisms. VReST meticulously traverses the reasoning landscape by establishing a search tree, where each node encapsulates a reasoning step, and each path delineates a comprehensive reasoning sequence. Our innovative multimodal Self-Reward mechanism assesses the quality of reasoning steps by integrating the utility of sub-questions, answer correctness, and the relevance of vision-language clues, all without the need for additional models. VReST surpasses current prompting methods and secures state-of-the-art performance across three multimodal mathematical reasoning benchmarks. Furthermore, it substantiates the efficacy of test-time scaling laws in multimodal tasks, offering a promising direction for future research.

1 Introduction

Chain-of-Thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023; Zhang et al., 2022) has been widely recognized as an effective technique for enhancing the performance of Large Language Models (LLMs) on complex reasoning tasks. Recently, OpenAI o1 (OpenAI, 2024) demonstrated the potential of generating ultra-long CoTs to achieve inference scaling laws.

Building on this progress, many studies (Zhang et al., 2023; Mitra et al., 2024; Shao et al., 2024; Zheng et al., 2023; Gao et al., 2024; Liu et al., 2024; Wu et al., 2024) have extended CoT prompting to Large Vision-Language Models (LVLMs), aiming to enhance their reasoning capabilities in multimodal tasks. While these methods show promise, they often generate limited intermediate reasoning steps and lack the ability to evaluate and refine the generated CoTs. Consequently, these approaches fail to fully unleash the reasoning potential of LVLMs, resulting in marginal improvements on challenging multimodal reasoning tasks (Zhang et al., 2025). As illustrated in Tables 1, 2, and 3, multimodal CoT reasoning underperforms direct question answering (Direct QA) on more complex visual mathematical tasks.

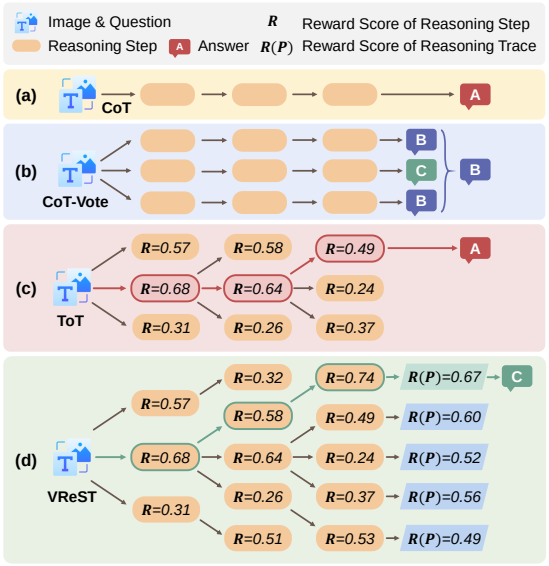


Figure 1: The difference between VReST and the previous multimodal CoT prompting methods. The methods in (a)(b)(c) obtain suboptimal solutions by a greedy algorithm, while VReST in (d) can fully explore the reasoning space to obtain the optimal solution.

To improve LVLM reasoning, a potential solution is to construct large LVLM reasoning datasets (Chen et al., 2024; Xu et al., 2023; Shao et al., 2024) and train LVLMs (Cheng et al., 2024; Guo et al., 2024; Zhang et al., 2024a). However, this approach is expensive and difficult to scale. Thus, we focus on developing training-free methods to enhance the reasoning ability of LVLMs.

Recent studies have shown that LLM with Monte

Carlo Tree Search (MCTS) (Hao et al., 2023; Zhang et al., 2024b; Jiang et al., 2024; Long, 2023; Yao et al., 2024) can effectively expand the reasoning space in a training-free manner, improving CoT generation. Based on these findings, we extend the MCTS algorithm to LVLM. A key component of any tree search algorithm is the reward function, which guides the model's exploration within the vast space of possible reasoning traces (Feng et al., 2023). To ensure a fair comparison with baseline methods, we avoid introducing additional models. Hence, we propose a multimodal Self-Reward mechanism that incorporates visual knowledge with textual clues.

To tackle the intricacies of complex vision tasks within LVLMs, we introduce **VReST**, a pioneering approach that Enhancing **Re**asoning in Large **V**ision-Language Models through **T**ree Search and **S**elf-Reward mechanism. Figure 1 shows the difference between VReST and existing multimodal CoT methods. VReST employs MCTS to systematically navigate the reasoning space, where nodes symbolize individual reasoning steps, and paths constitute complete reasoning trajectories. By recursively identifying nodes with high confidence, VReST dynamically crafts reasoning steps and fosters diversity by modulating the temperature of LVLM generation, thus enriching the exploration of the reasoning space. Based on prior work (Hao et al., 2023), we present a multimodal Self-Reward mechanism that appraises the merit of reasoning steps. It considers sub-question utility, final answer correctness, and vision-language clues. Inspired by (Lightman et al., 2023), our mechanism assigns reward values to each node.

Finally, VReST expands, evaluates, and backpropagates reasoning traces in each iteration, thereby refining the search tree by updating node statistics. The optimal reasoning trace is selected based on the aggregate reward, with the final answer being extracted from the terminal node. Experiments show that VReST outperforms existing prompting methods on three visual reasoning datasets. Moreover, as shown in Section 4.7, the performance gain of our approach becomes more pronounced with increasing iterations of MCTS, surpassing other prompting methods, and demonstrating better multimodal test-time scaling. Our approach offers a promising direction for training-free methods to enhance LVLM reasoning.

Our main contributions are as follows:

- We introduce a training-free approach that uses MCTS to enhance the depth and quality of reasoning in LVLMs.
- We propose a Self-Reward mechanism incorporating visual information to evaluate reasoning traces.
- We achieve SOTA performance on three multimodal mathematical reasoning datasets, outperforming existing prompting methods.
- We demonstrate that VReST exhibits a better test-time scaling law in multimodal tasks.

2 Related Work

2.1 CoT for Large Vision-Language Models

Large Vision-Language Models (LVLMs) demonstrate remarkable abilities in integrating visual and linguistic information (Li et al., 2024; Peng et al., 2024), but face challenges in tasks requiring complex reasoning or multi-hop inferences (Lu et al., 2023; Wang et al., 2024a,c; Zhao et al., 2024; Chen et al., 2024). Extending the Chain of Thought (CoT) paradigm (Kojima et al., 2022; Zhang et al., 2022) to the multimodal domain offers a promising direction. While many approaches enhance the CoT reasoning abilities of LVLMs through extensive training (Xu et al., 2023; Shao et al., 2024; Cheng et al., 2024; Guo et al., 2024), optimizing reasoning traces provides a viable training-free alternative. Initial effort adopts a two-stage reasoning method (Zhang et al., 2023) where rationales precede the final answer to enable step-by-step inference. Subsequent advancements augment reasoning steps with precise visual details, such as scene graphs (Mitra et al., 2024) and related image regions (Shao et al., 2024). To better understand textual information, DDCoT (Zheng et al., 2023) decomposes questions into sub-questions, and utilize sub-answers to construct reasoning steps. Cantor (Gao et al., 2024) further improves this approach by framing LVLMs as multifaceted experts for multi-step reasoning.

However, these methods struggle with complex questions due to limited reasoning steps and lack of feedback to refine traces. VReST addresses these issues with a tree search for extended reasoning and reward evaluation for optimal solutions.

2.2 Tree-based Reasoning with LLMs

Tree-based reasoning methods enhance performance by increasing computational costs to explore diverse solution spaces (Jiang et al., 2024). Self-Consistency (Wang et al., 2022) improves accuracy

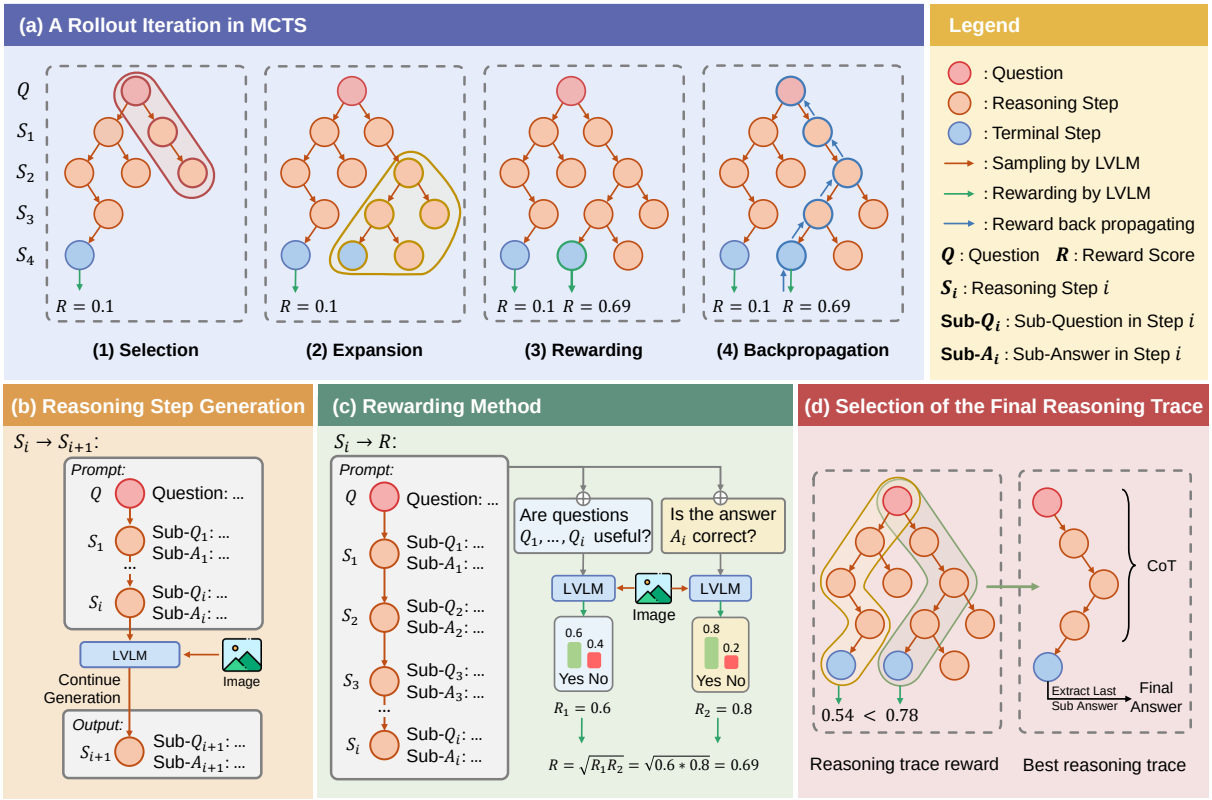


Figure 2: The framework of VReST. (a) Illustrates the MCTS rollout iteration process, including Selection, Expansion, Rewarding, and Backpropagation steps. (b) Depicts the generation of new reasoning steps using LVLM based on the constructed prompt. (c) Shows the Self-Rewarding mechanism for calculating the reward of new reasoning steps, considering both the usefulness of sub-questions and the correctness of the last answer. (d) Describes the Best-Trace strategy of the final reasoning trace selection.

by sampling multiple reasoning traces, while Tree of Thoughts (ToT) (Long, 2023; Yao et al., 2024) use heuristic methods to select optimal steps but often converges to locally optimal solutions. Breadth-First Search(BFS) (Yao et al., 2024) identifies globally optimal reasoning traces by exploring the entire space. Monte Carlo Tree Search (MCTS) (Hao et al., 2023; Zhang et al., 2024b) further integrates rewarding and backpropagation mechanisms, quantifying each inference trace across multiple iterations to identify the globally optimal solution. Despite their potential, tree-based reasoning methods have rarely been applied to multimodal reasoning tasks. Our framework incorporates visual information into reasoning steps and, to the best of our knowledge, is the first to employ MCTS for multimodal CoT reasoning.

3 Method

As shown in Figure 2, our approach combines Monte Carlo Tree Search (MCTS) with Large Vision-Language Model (LVLM) to generate step-by-step reasoning traces and evaluate them using a Self-Rewarding mechanism. Below, we detail the problem formulation (3.1), the MCTS framework with a Self-Reward mechanism (3.2), as well as the final reasoning trace selection method (3.3).

3.1 Problem Formulation

Given a question Q and an image I, our goal is to find the optimal reasoning trace $\mathcal{P}^*$ that leads to the correct answer A. Each reasoning trace $\mathcal{P}$ consists of an original question and a sequence of reasoning steps: $\{Q, S_1, S_2, ..., S_n\}$, where each step S_i contains a sub-question Q_i and its corresponding sub-answer A_i.

3.2 Monte Carlo Tree Search Framework

In Figure 2(a), we employ MCTS to explore the reasoning space systematically. Each node in the search tree represents a reasoning step S_i, and edges represent the transitions between steps. The rollout iteration in MCTS involves four steps: Selection, Expansion, Rewarding, and Backpropagation. These steps are iteratively performed K times to explore the reasoning space and refine the search

tree. The experiments in section 4.7 show that VReST efficiently utilizes additional iterations to refine its reasoning traces, and exhibits a test-time scaling law on multimodal reasoning tasks.

3.2.1 Selection

In Figure 2(a)(1), we select a path in the search tree. Starting from the root node (original question Q), we recursively select child nodes according to the Upper Confidence Bound applied to Trees (UCT) algorithm (Kocsis and Szepesvári, 2006), which selects a node v by balancing exploration and exploitation:

$$UCT(v) = R(v) + c\sqrt{\frac{\ln N(p(v))}{N(v)}}, \quad (1)$$

where $R(v)$ is the reward value of node v, $N(v)$ is the visit count, $p(v)$ is the parent node, and c is the exploration constant. The child node with the highest UCT value is recursively selected until a leaf node is reached.

3.2.2 Expansion

We generate new reasoning steps for the selected path S_t using LVLM. As shown in Figure 2(b), the prompt for generation is constructed as:

$$\mathcal{P}_{t-1} = [Q, S_1, \ldots, S_{t-1}]. \quad (2)$$

Based on the prompt $\mathcal{P}_{t-1}$, LVLMs are prompted to generate w distinct reasoning steps S_t by increasing the temperature parameter of LVLMs:

$$\{S_{t,j} | j = 1, \ldots, w\} = \text{LVLM}(\mathcal{P}_{t-1}, I), \quad (3)$$

where w is the width of the tree.

Subsequently, the initial reward value of each child node is obtained using the Self-Reward mechanism described in Section 3.2.3. Then, we select the child node with the highest reward:

$$S_{t,\text{selected}} = \arg\max_j R(S_{t,j}), \quad (4)$$

where $R(S_{t,j})$ denotes the reward value for the j-th child node $S_{t,j}$. The selected node $S_{t,\text{selected}}$ becomes the current node in the reasoning trace, and the generation process continues to generate S_{t+1} according to Equations (2)(3)(4).

As shown in Figure 2(a)(2), this process continues iteratively until either a terminal node is reached or the maximum depth D_{max} of the tree is achieved. As shown in the prompt in Section D.1, when the sub-question generated by LVLM contains the span "Now we can answer the question", the node is considered to be a terminal node. In the case that the terminal node is reached, we stop the generation process and backpropagate the reward values as described in Section 3.2.4.

3.2.3 Rewarding

We introduce a Self-Rewarding mechanism to calculate the reward value of the new reasoning step S_t using two criteria: (1) Usefulness of all the sub-questions on the reasoning trace. (2) Correctness of the last answer on the reasoning trace.

First, as shown in Figure 2(c), we concatenate each reasoning step prior to S_t on the selected reasoning trace to construct the Rewarding prompt:

$$\mathcal{P}_t = [Q, S_1, \ldots, S_t]. \quad (5)$$

Then, we calculate the usefulness of all the sub-questions R_1 and the correctness of the last answer R_2, respectively, and then calculate their geometric mean as the reward value R of reasoning step S_t:

$$\begin{aligned} R_1 &= P(\text{"Yes"} | [\mathcal{P}_t, \mathcal{P}_Q], I), \\ R_2 &= P(\text{"Yes"} | [\mathcal{P}_t, \mathcal{P}_A], I), \\ R &= \sqrt{R_1 R_2}, \end{aligned} \quad (6)$$

where $P(\text{"Yes"}|\cdot)$ represents the probability that the first token generated by LVLM is "Yes". $\mathcal{P}_Q$ is "Are questions $Q_1, \ldots, Q_t$ useful?". $\mathcal{P}_A$ is "Is the answer A_t correct?".

3.2.4 Backpropagation

As shown in Figure 2(a)(4), when a terminal node S_T is reached, the reward values of each node are backpropagated through all nodes in the selected path, where the T is the number of reasoning steps in the selected path. For each node S_t in the path, where $t = 1, \ldots, T$, we update its statistics by aggregating the rewards in all future steps of S_t:

$$\begin{aligned} R(S_t) &= Avg(\{R(S_i)\}_{i=t}^{T}), \\ N(S_t) &= N(S_t) + 1. \end{aligned} \quad (7)$$

3.3 Final Reasoning Trace Selection

After completing K MCTS iterations, we select the final reasoning trace $\mathcal{P}^*$ based on the trace rewards. There are three ways for the reasoning trace selection.

Greedy Trace. Starting from root node Q, we select the reasoning trace $\mathcal{P}^*$ by greedily choosing the node with the highest reward at each step.

Methods	FQA	GPS	MWP	TQA	VQA	ALG	ARI	GEO	LOG	NUM	SCI	STA	ALL
QA	60.59	48.56	60.75	56.96	50.28	49.11	52.69	46.03	16.22	34.03	59.84	67.44	55.70
CoT	63.57	40.87	56.99	62.03	48.04	45.91	50.42	42.68	18.92	40.28	59.02	70.43	54.60
CoT-Vote	**70.63**	48.08	69.89	63.92	56.98	51.60	60.34	50.63	10.81	51.39	60.66	**79.07**	62.30
Best-of-N	67.66	44.71	59.68	58.86	54.75	48.75	54.96	46.03	13.51	43.06	56.56	75.42	57.70
Cantor	63.57	48.08	62.90	61.39	56.42	50.89	55.81	49.37	21.62	45.83	60.66	70.43	58.60
ToT	66.54	53.37	63.44	61.39	54.19	54.80	55.24	54.39	13.51	43.75	57.38	74.09	60.20
VReST	68.03	**56.73**	72.04	**67.09**	58.10	**59.43**	62.61	**58.16**	**29.73**	50.69	67.21	75.75	64.50
VReST-Vote	69.14	51.44	**75.81**	66.46	**64.25**	54.45	**67.42**	53.56	27.03	**60.42**	**68.03**	77.74	**65.40**

Table 1: Accuracy (%) on the testmini set of MathVista, where bold indicates the best results, underlines indicate the second-best. Task types: FQA: figure question answering, GPS: geometry problem solving, MWP: math word problem, TQA: textbook question answering, VQA: visual question answering. Mathematical reasoning types: ALG: algebraic reasoning, ARI: arithmetic reasoning, GEO: geometry reasoning, LOG: logical reasoning, NUM: numeric commonsense, SCI: scientific reasoning, STA: statistical reasoning. ALL: overall accuracy.

Best Trace. As shown in Figure 2(d), we calculate the reward value for each trace in the tree:

$$R(\mathcal{P}) = Avg(\{R(S_t)|S_t \in \mathcal{P}, t = 1, \ldots, T\}). \tag{8}$$

And then select the trace with the highest value:

$$\mathcal{P}^* = \arg\max_{\mathcal{P}} R(\mathcal{P}), \tag{9}$$

where $R(\mathcal{P})$ denotes the reward value for the trace $\mathcal{P}$. **Best-Trace** is written **VReST** in Tables 1, 2, 3.

Trace Vote. Similar to CoT-Vote, after calculating the reward of all the reasoning traces by Equation (8), we select the n with the highest reward value. **Trace-Vote** is written **VReST-Vote** in Tables 1, 2, 3.

For the Greedy Trace and Best Trace, the final answer A_T^* is extracted from the terminal node S_T^* of the selected trace $\mathcal{P}^*$. For the Trace Vote, the final answer A_T^* is obtained by extracting the majority of the answers from the n selected traces. In practice, we observe that the Best Trace and Trace Vote strategies usually yield the best results.

4 Experiments

4.1 Datasets

We evaluate our approach on three visual reasoning datasets: **MathVista** (Lu et al., 2023), **MathVision** (Wang et al., 2024a) and **CharXiv** (Wang et al., 2024c). All datasets are evaluated using answer accuracy. See Appendix A for more details on the datasets.

4.2 Models

The LVLM used in this paper is Qwen2-VL-7B-Instruct (Wang et al., 2024b). The LVLM is utilized in three components: (1) Generating reasoning steps during expansion. (2) Calculation of R_1 in Rewarding method. (3) Calculation of R_2 in Rewarding method. The temperature of LVLM is 0.7, the top_p is 0.95.

The text-only LLM used in this paper is Qwen2.5-7B-Instruct (Yang et al., 2024). The text-only LLM is utilized in two components: (1) Evaluating whether the final answers and golden answers are consistent. (2) Replacing LVLM in the VReST in ablation experiments in Section 4.6. The temperature of text-only LLM is 0.7, the top_p is 0.95.

4.3 Baselines

We compare VReST with six baselines: **Question Answering (QA)**, **Chain of Thought (CoT)** (Kojima et al., 2022), **CoT-Vote** (Wang et al., 2022), **Best-of-N** (Lightman et al., 2023), **Cantor** (Gao et al., 2024), **Tree of Thought (ToT)** (Yao et al., 2024). We control the parameters of the baseline methods to be consistent with VREST, doing our best to maintain a fair comparison. See Appendix B for more details on baselines.

4.4 Implementation Details of VReST

For each MCTS iteration, we maintain a maximum depth of $D_{max} = 8$ steps and perform $K = 10$ total iterations to ensure adequate exploration of the reasoning space. The exploration constant $c = 1$ in the UCT formula is set to balance exploration and exploitation during the search process. The width of the tree is $w = 5$. In the **VReST-Vote**, the selected number of reasoning traces is $n = K$. The prompts are shown in Appendix D.

4.5 Main Results

MathVista. The results presented in Table 1 clearly highlight the superior performance of VReST and VReST-Vote across various mathematical and visual reasoning tasks on the testmini sub-

Methods	ALG	AnaG	Ari	CombG	Comb	Cnt	DescG	GrphT	Log	Angle	Area	Len	SolG	Stat	Topo	TransG	ALL
QA	15.79	15.79	10.53	21.05	0.00	5.26	5.26	21.05	15.79	**57.89**	15.79	**36.84**	15.79	15.79	15.79	26.32	18.42
CoT	15.79	10.53	15.79	10.53	15.79	10.53	26.32	15.79	15.79	10.53	0.00	10.53	15.79	26.32	21.05	10.53	14.47
CoT-Vote	0.00	26.32	**21.05**	15.79	**42.11**	**26.32**	5.26	26.32	15.79	21.05	31.58	10.53	21.05	31.58	31.58	21.05	21.71
Best-of-N	5.26	31.58	0.00	21.05	21.05	**26.32**	26.32	15.79	15.79	36.84	26.32	21.05	10.53	21.05	15.79	10.53	19.08
Cantor	5.26	21.05	10.53	15.79	15.79	10.53	0.00	10.53	21.05	15.79	10.53	0.00	5.26	15.79	5.26	15.79	11.18
ToT	**21.05**	26.32	15.79	21.05	21.05	15.79	15.79	15.79	5.26	31.58	**36.84**	21.05	15.79	**42.11**	10.53	10.53	20.39
VReST	**21.05**	31.58	**21.05**	21.05	15.79	10.53	10.53	**42.11**	**42.11**	15.79	**36.84**	10.53	**26.32**	31.58	**52.63**	**36.84**	26.64
VReST-Vote	10.53	**42.11**	15.79	**31.58**	21.05	21.05	**36.84**	36.84	26.32	42.11	26.32	31.58	15.79	31.58	36.84	26.32	**28.29**

Table 2: Accuracy scores (%) on the testmini subset of MATH-Vision. Alg: algebra, AnaG: analytic geometry, Ari: arithmetic, CombG: combinatorial geometry, Comb: combinatorics, Cnt: counting, DescG: descriptive geometry, GrphT: graph theory, Log: logic, Angle: metric geometry - angle, Area: metric geometry - area, Len: metric geometry-length, SolG: solid geometry, Stat: statistics, Topo: topology, TransG: transformation geometry.

set of MathVista. VReST achieves notable success, outperforming other methods in tasks such as MWP with 72.04%, SCI with 67.21%, and STA with 75.75%. Additionally, the VReST-Vote method further elevates accuracy, particularly in tasks such as MWP (75.81%), VQA (64.25%), and NUM (60.42%), by aggregating multiple reasoning traces through a voting mechanism. This reflects VReST's robust ability to handle complex reasoning challenges that require logical, numerical, and scientific understanding. Its strength lies in the combination of MCTS for systematic exploration of reasoning traces and the Self-Reward mechanism, which dynamically evaluates reasoning steps based on sub-question utility, answer correctness and visual information. This allows VReST to refine its reasoning traces over time, enhancing performance in a diverse set of tasks

MathVision. In Table 2, we evaluate various methods on the testmini subset of the MATH-Vision dataset, which includes a range of mathematical and visual reasoning tasks. VReST achieves an overall accuracy of 26.64%, outperforming baseline and competitive methods, with notable results in GrphT (42.11%), Log (42.11%), and Topo (52.63%), outperforming other methods such as QA, CoT, and ToT in these tasks, showcasing its ability to handle complex geometric reasoning. The VReST-Vote method further improves this to 28.29%, excelling in tasks like AnaG (42.11%), DescG (36.84%), and Angle (42.11%). This demonstrates the effectiveness of the voting mechanism in aggregating diverse reasoning traces, leading to more reliable and accurate solutions. The integration of MCTS and the Self-Reward mechanism in VReST allows it to effectively explore reasoning traces and dynamically adjust to improve performance, particularly in challenging areas like combinatorics and graph theory.

CharXiv. The results presented in Table 3 on the validation set of the CharXiv dataset clearly highlight the superiority of VReST and VReST-Vote across various domains, particularly in tasks involving complex visual reasoning and interpretation of charts and graphs. VReST achieves an overall accuracy of 33.10%, outperforming baseline methods, with notable results in Text in General (54.55%), Num in Chart (33.62%), and Mathematics (40.74%). VReST-Vote improves this to 38.10%, with strong performances in Text in General (61.62%), Num in Chart (39.22%), and Electrical Engineering and Systems Science (45.38%), demonstrating the effectiveness of the voting mechanism in aggregating diverse reasoning traces. The results indicate that VReST-Vote not only achieves superior performance in individual tasks but also significantly outperforms other methods across a wide range of subjects, highlighting its robustness in addressing the challenges of complex visual reasoning in the CharXiv dataset.

4.6 Ablation Results

The importance of visual information. To illustrate the importance of visual information, we conducted ablation experiments shown in Figure 3a. As described in Section 4.2, the LVLM is utilized in three components. We performed ablation experiments by replacing LVLM with text-only LLM in each component separately. The study evaluates different configurations of visual and text-only components across three datasets: MathVista, MathVision, and CharXiv. The configuration where all components (reasoning generation, R1, and R2 reward computation) use LVLM achieves the highest performance across all datasets. When visual components are partially replaced with text-only components, the performance drops significantly. The ablation study clearly demonstrates that

Methods	Text in Chart	Text in General	Num in Chart	Num in General	CS	EC	EESS	MATH	PHY	QB	QF	STA	ALL
QA	31.82	38.38	28.45	22.27	33.33	30.43	31.93	29.63	35.43	25.40	21.55	27.43	29.50
CoT	29.09	40.40	26.72	18.78	21.43	27.54	32.77	29.63	26.77	23.81	23.28	33.63	27.30
CoT-Vote	32.95	45.45	28.88	22.71	26.98	28.99	33.61	30.37	**39.37**	29.37	25.86	32.74	30.90
Best-of-N	34.09	48.48	28.02	24.02	33.33	30.43	30.25	35.56	38.58	**31.75**	24.14	29.20	31.80
Cantor	27.73	43.43	30.60	23.58	26.19	27.54	27.73	31.11	37.01	24.60	30.17	27.43	29.00
ToT	34.09	45.45	33.62	20.96	30.95	26.81	36.97	31.85	35.43	29.37	26.72	**39.82**	32.10
VReST	33.64	54.55	33.62	22.27	30.95	31.16	41.18	40.74	33.86	26.98	30.17	29.20	33.10
VReST-Vote	**37.95**	**61.62**	**39.22**	**27.07**	**37.30**	**38.41**	**45.38**	**43.70**	38.58	**31.75**	**36.21**	32.74	**38.10**

Table 3: Accuracy scores (%) on the Validation set of CharXiv. CS: Computer Science, EC: Economics, EESS: Electrical Engineering and Systems Science, MATH: Mathematics, PHY: Physics, QB: Quantitative Biology, QF: Quantitative Finance, STA: Statistics.

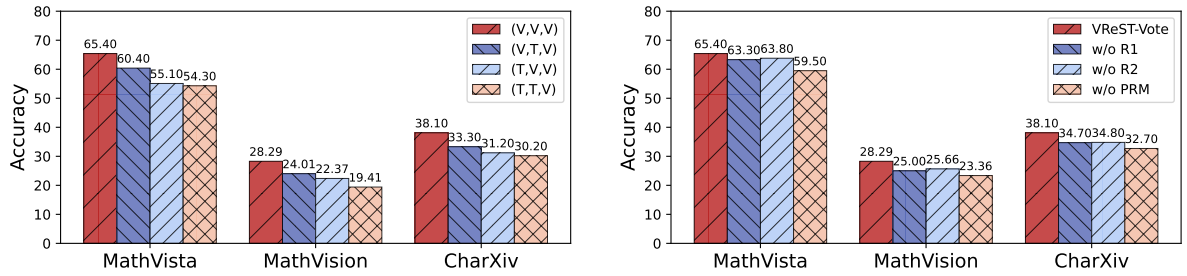


Figure 3: (a) Ablation results of different configurations of visual and text-only components. (V, V, V) represents using LVLM for all components (reasoning generation, R1, and R2 rewarding), while T denotes using text-only LLM. (b) Ablation results from different reward methods. `w/o R1` and `w/o R2` denote R1 or R2 is omitted, respectively. `w/o PRM` indicates that the Process Reward Model is no longer employed.

visual information is indispensable for LVLM to solve complex visual reasoning tasks. Our method, VReST, leverages Large Vision-Language Models (LVLM) to integrate visual and textual information seamlessly, enabling the generation of accurate and reliable reasoning traces. Specifically, the Self-Rewarding mechanism in VReST relies on both visual and textual information to evaluate reasoning traces effectively. Without visual input, the model loses the ability to make informed decisions, especially in tasks that involve interpreting visual elements such as charts, graphs, and geometric figures. This is particularly evident in datasets like MathVision and CharXiv, where visual reasoning plays a central role.

The importance of reward method. To demonstrate the effectiveness of our Self-Rewarding mechanism, we conducted ablation experiments as shown in Figure 3b. Specifically, `w/o R1` and `w/o R2` denote the scenarios where R1 or R2 is omitted during the calculation of the reward value, respectively. `w/o PRM` indicates that the Process Reward Model is no longer employed; instead, only the reward value of the terminal node is computed, while the reward value of non-terminal nodes is uniformly set to 0.5. In this case, the reward of non-terminal nodes is updated solely through the back-propagation mechanism. The ablation study clearly demonstrates that the Self-Rewarding mechanism in VReST-Vote is indispensable for achieving high accuracy in complex reasoning tasks. The R1 reward ensures that each reasoning step is evaluated and guided toward correctness, while the R2 reward evaluates the final answer to ensure the overall trace is accurate. The Process Reward Method (PRM) plays a crucial role in assigning intermediate rewards to non-terminal nodes, guiding the reasoning process effectively. Omitting any of these components leads to a significant performance drop, highlighting the importance of a comprehensive reward mechanism.

The importance of selection method. We analyze the results of different selection methods for final trace evaluation, as presented in Table 4. As described in Section 3.3, there are three methods for the selection of the final trace and evaluation of the final answer: Greedy-Trace, Best-Trace, and Trace-Vote. The study evaluates three methods across three datasets: MathVista, MathVision, and CharXiv. The results of the ablation

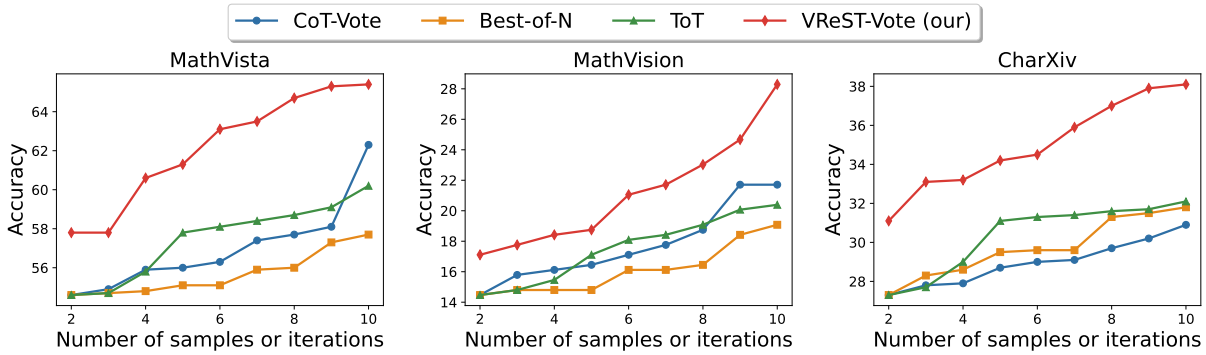


Figure 4: The impact of the number of samples or iterations. It shows that our VReST exhibits a better test-time scaling law than other SOTA methods in multimodal reasoning tasks.

Methods	MathVista	MathVision	CharXiv
Trace-Vote	**65.40**	**28.29**	**38.10**
Best-Trace	64.50	26.64	33.10
Greedy-Trace	60.00	23.03	31.30

Table 4: Results of different selection methods.

study on selection methods demonstrate that the Trace-Vote method is the most effective for final trace evaluation. By leveraging a voting mechanism to aggregate multiple high-reward reasoning traces, Trace-Vote achieves superior performance across all datasets. It effectively mitigates the risk of selecting a suboptimal trace by considering a broader range of potential solutions. In contrast, the Greedy-Trace method relies on a single trace selection strategy, suffering from a significant performance drop. This indicates that a greedy approach may not fully capture the complexity of the reasoning process, especially in tasks that require deep visual and logical reasoning. The Best-Trace method, while performing better than Greedy-Trace, is still outperformed by Trace-Vote. This suggests that selecting the single best trace, although effective, does not fully exploit the potential of multiple high-reward traces. The voting mechanism in Trace-Vote provides a more robust and reliable way to determine the final answer, especially in complex tasks that involve multiple reasoning steps.

4.7 Multimodal Test-Time Scaling Law

To investigate the impact of different methods on the number of samples or iterations, we conducted hyperparameter experiments as shown in Figure 4 by controlling the number of samples in each method. The study evaluates the performance of CoT-Vote, Best-of-N, ToT, and VReST-Vote across three datasets: MathVista, MathVision, and CharXiv. The x-axis of Figure 4 corresponds to different hyperparameters across various baseline methods. Specifically, in CoT-Vote, the x-axis represents the number of votes n. In Best-of-N, the x-axis denotes the number of sampled reasoning traces n. In ToT, the x-axis represents the width of the tree w. In VReST-Vote, the x-axis corresponds to the number of iterations for MCTS K.

It can be observed that VReST-Vote consistently outperforms the baselines across all numbers of samples or iterations. The superior performance of VReST-Vote can be attributed to its Monte Carlo Tree Search (MCTS) algorithm, which efficiently explores the search space and converges to optimal solutions with relatively fewer iterations. Moreover, VReST-Vote shows a more significant performance improvement than the baselines as the number of iterations increases, indicating that it efficiently utilizes additional iterations to refine its reasoning traces. This proves that our method exhibits a better test-time scaling law on multimodal reasoning tasks.

5 Conclusion

In this paper, we presented VReST, a novel training-free approach that enhances reasoning capabilities in Large Vision-Language Models through Monte Carlo Tree Search and Self-Reward mechanism. Through extensive experiments on three challenging multimodal mathematical reasoning datasets, VReST significantly outperformed existing prompting methods and achieved state-of-the-art performance. Furthermore, we validate test-time scaling laws' applicability to multimodal tasks, offering a promising direction to improving LVLM performance for future research.

Limitations

Although our results already outperform baselines overall, our work still suffers from the following limitations.

Self-Reward Mechanism To ensure a fair comparison with baseline methods, we designed the self-reward mechanism to use the LVLM itself for reward scoring, without introducing additional models. This approach aligns with the training-free nature of our method, enabling quick deployment without the need for training a separate reward model. However, this mechanism heavily relies on the LVLM's own judgments to evaluate the quality of reasoning traces. As a result, there is a risk that model biases or errors could propagate through the reward process, potentially affecting the accuracy and reliability of the reasoning process. Future work could involve training an additional reward model to assist the LVLM's reasoning process, helping to mitigate potential biases and improve the accuracy of the reward signal.

Computational Cost The MCTS approach relies on multiple iterations and extensive tree exploration, resulting in significant computational overhead compared to current prompting methods. This increased cost may limit the scalability of VReST for large-scale applications. In future work, we aim to address this by incorporating pruning strategies or early stopping techniques within the tree search process, which could help reduce the computational burden while maintaining performance.

Model Dependency Currently, we have only evaluated the effectiveness of VReST on the Qwen2-VL-7B-Instruct model. Although this model demonstrates the benefits of our approach, the effectiveness of VReST may vary across different LVLMs, especially models with different architectures, scales, or training regimens. In future work, further experimentation on a wider range of LVLMs will be essential to determine the generalizability of our approach.

Dataset Dependency Our experiments primarily focus on a limited set of visual reasoning datasets. While VReST shows promising results on these datasets, its performance on other datasets with different characteristics, such as those involving diverse types of reasoning or tasks outside visual reasoning, remains unexplored. Expanding our evaluation to a broader set of datasets will help assess the robustness and versatility of VReST across different multimodal tasks.

References

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. arXiv preprint arXiv:2405.16473.

Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. 2024. Vision-language models can self-improve reasoning via reflection. arXiv preprint arXiv:2411.00855.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. arXiv preprint arXiv:2309.17179.

Timin Gao, Peixian Chen, Mengdan Zhang, Chaoyou Fu, Yunhang Shen, Yan Zhang, Shengchuan Zhang, Xiawu Zheng, Xing Sun, Liujuan Cao, et al. 2024. Cantor: Inspiring multimodal chain-of-thought of mllm. In Proceedings of the 32nd ACM International Conference on Multimedia, pages 9096–9105.

Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhu Chen, and Xiang Yue. 2024. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. arXiv preprint arXiv:2412.05237.

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

Jinhao Jiang, Zhipeng Chen, Yingqian Min, Jie Chen, Xiaoxue Cheng, Jiapeng Wang, Yiru Tang, Haoxiang Sun, Jia Deng, Wayne Xin Zhao, et al. 2024. Technical report: Enhancing llm reasoning with reward-guided tree search. arXiv preprint arXiv:2411.11694.

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.

Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. 2024. How to configure good in-context sequence for visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 26710–26720.

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

Mengsha Liu, Daoyuan Chen, Yaliang Li, Guian Fang, and Ying Shen. 2024. Chartthinker: A contextual chain-of-thought approach to optimized chart summarization. arXiv preprint arXiv:2403.11236.

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

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14420–14431.

OpenAI. 2024. Introducing openai o1-preview. Accessed: 2024-12-13.

Yingzhe Peng, Xinting Hu, Jiawei Peng, Xin Geng, Xu Yang, et al. 2024. Live: Learnable in-context vector for visual question answering. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024a. Measuring multimodal mathematical reasoning with math-vision dataset. arXiv preprint arXiv:2402.14804.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.

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.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. 2024c. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. arXiv preprint arXiv:2406.18521.

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

Yifan Wu, Lutao Yan, Yuyu Luo, Yunhai Wang, and Nan Tang. 2024. Evaluating task-based effectiveness of mllms on charts. arXiv preprint arXiv:2405.07001.

Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2023. Chartbench: A benchmark for complex visual reasoning in charts. arXiv preprint arXiv:2312.15915.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. arXiv preprint arXiv:2407.10671.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 36.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. arXiv preprint arXiv:2406.03816.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, et al. 2024b. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. arXiv preprint arXiv:2410.02884.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. 2025. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In European Conference on Computer Vision, pages 169–186. Springer.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. arXiv preprint arXiv:2210.03493.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. arXiv preprint arXiv:2302.00923.

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. 2024. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. arXiv preprint arXiv:2406.12742.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. 2023. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. Advances in Neural Information Processing Systems, 36:5168–5191.

A Datasets

We evaluate our approach on three visual reasoning datasets. The details are given below:

MathVista (Lu et al., 2023) is a comprehensive benchmark dataset designed to evaluate the mathematical reasoning capabilities of foundation models in visual contexts. It consists of 6,141 examples derived from 28 existing multimodal datasets and 3 newly created datasets: IQTest, FunctionQA, and PaperQA. These datasets address the need for evaluating logical reasoning on puzzle test figures, algebraic reasoning over functional plots, and scientific reasoning with academic paper figures, respectively. In this paper, we used Mathvista testmini, which includes 1000 samples.

MathVision (Wang et al., 2024a) is a meticulously curated collection of 3,040 high-quality mathematical problems with visual contexts, sourced from real math competitions such as Math Kangaroo, AMC, and AIME. Spanning 16 distinct mathematical disciplines and graded across 5 levels of difficulty, it provides a comprehensive benchmark for evaluating the multimodal mathematical reasoning capabilities of large multimodal models (LMMs). The dataset emphasizes both visual perception and mathematical reasoning, covering topics like algebra, topology, and graph theory, and includes both multiple-choice and free-form questions. In this paper, we used MathVision testmini, which includes 304 samples.

CharXiv (Wang et al., 2024c) is a comprehensive evaluation suite designed to rigorously assess the chart understanding capabilities of Multimodal Large Language Models. Comprising 2,323 natural, diverse, and challenging charts sourced from arXiv scientific papers, CharXiv addresses the limitations of existing datasets that often rely on oversimplified, homogeneous charts and template-based questions, leading to an over-optimistic assessment of model performance. The dataset includes two types of questions: descriptive questions that focus on extracting basic chart elements and reasoning questions that require synthesizing complex visual and numerical information across charts. To better evaluate the model's ability to solve complex problems, we use all reasoning questions from the validation set of CharXiv, which includes 1,000 samples.

B Baselines

We compare VReST with six baseline methods. We control the parameters of the baseline method to be consistent with VREST, doing our best to maintain a fair comparison.

Question Answering (QA). It is a straightforward prompting method where the model is given a question and image and expected to generate a direct answer without any intermediate reasoning steps.

Chain of Thought (CoT) (Kojima et al., 2022). It is a prompting technique that guides the model to break down complex questions into a series of simpler sub-questions and solve them sequentially. In this paper, we implement zero-shot CoT by explicitly asking the model to decompose the original question into sub-questions. To ensure a fair comparison, for the generation of sub-questions and answers in CoT, we use the same prompt as shown in Appendix D.1.

CoT-Vote (Wang et al., 2022). It extends the CoT approach by generating multiple reasoning chains and selecting the most frequent answer from among them. This method is also known as Self-Consistency. In this paper, the number of votes in CoT-Vote is $n = 10$.

Best-of-N (Lightman et al., 2023). It is an alternative to CoT-Vote, where the reasoning trace with the highest reward value is selected from multiple reasoning traces as the final answer. We calculate the reward value for the last step of each reasoning trace in CoT-Vote using the rewarding method described in Section 3.2.3, and then select the one with the highest value. In this paper, the number of reasoning traces in Best-of-N is $n = 10$.

Cantor (Gao et al., 2024). It uses an LVLM as a decision maker to break down the question into different parts, which are then assigned to different experts (also LVLMs) for processing, and finally the results of each expert are summarized to obtain the final answer.

Tree of Thought (ToT) (Yao et al., 2024). We reproduce the same method as in ToT's paper. When generating each reasoning step, we sample multiple different child nodes, and then calculate the reward value of each child node through the rewarding method in Section 3.2.3. The node with the highest value is then iteratively selected in a greedy decoding-like manner until a terminating node is generated. To ensure a fair comparison, for the generation of sub-questions and answers in ToT,

we use the same prompt as shown in Appendix D.1. The width of the tree in ToT is $w = 10$, and the maximum depth in ToT is $D_{max} = 8$.

C Algorithm

Algorithm 1 below presents the algorithm used in our VReST framework.

D Prompt Templates

To ensure a fair comparison, for the generation of reasoning steps in VReST, CoT, CoT-Vote, and ToT, we use the same prompts as in the previous work (Hao et al., 2023), as shown in Appendix D.1. For the prompts in Appendix D.1, D.2, and D.3, the samples in the prompts are only used to guide the LVLM in generating content in the expected format, and no multimodal samples are included in the prompts. Therefore, we consider the method in this paper to be a zero-shot prompting technique. For all methods in this paper, we use the prompt template in Appendix D.4 to judge whether the final answer is correct or not.

D.1 Reasoning Step Generation

As shown in the Prompt Template of Reasoning Step Generation, we input $k-1$ sub-questions and corresponding answers and let LVLM continue to generate the k-th sub-question and corresponding answer. Model-generated content is annotated in blue.

D.2 R1 Rewarding

As shown in the Prompt Template of Calculating Usefulness of All the Sub-questions, we feed the current sub-questions and the latest sub-question into LVLM and let it judge whether the new sub-question is useful or not. Model-generated content is annotated in blue.

D.3 R2 Rewarding

As shown in the Prompt Template of Calculating Correctness of the last Answer, we feed all the current sub-questions and their corresponding answers into LVLM and let it judge whether the last answer is correct or not. Model-generated content is annotated in blue.

D.4 Answer Evaluation

As shown in the Prompt Template for answer evaluation, we feed the predicted answer together with the ground truth into the text-only LLM and let it judge whether the predicted answer is correct or not.

E Case Study

Figure 5 evaluates the capability of VReST in solving a series of multimodal reasoning problems involving numerical and visual patterns. The tasks test the ability of reasoning frameworks to interpret relationships, verify intermediate steps, and derive accurate conclusions across diverse scenarios.

To address these problems, we compare three frameworks: CoT, ToT, and our proposed VReST. In Case 1, which involves summing corresponding values from a grid to determine a missing number, CoT incorrectly calculates 10+13=23, failing to verify intermediate results like 6+11=17. ToT improves by adopting a tree structure but still misjudges node selection, concluding an incorrect answer of 11. In contrast, VReST uses MCTS to explore alternatives systematically, accurately deriving 8+7=15 as the solution.

In Case 2, which requires identifying unique digits in a drawing, CoT lists visible digits as 0,5,3 but overlooks others like 2,8,9, resulting in an incomplete answer of 3. ToT detects additional digits but fails to verify their uniqueness, producing an erroneous total of 6. VReST, leveraging visual clues such as digits on the face and feet, systematically identifies all unique digits 0,5,3,2,8,9, arriving at the correct answer of 6.

In Case 3, which involves solving a grid of algebraic equations, CoT's linear reasoning misses critical steps, leading to an incorrect answer of 7. ToT applies tree-based reasoning but inadequately propagates constraints, yielding 11 as the result. VReST, however, integrates equations like 4+7+?=11 and verifies intermediate solutions, correctly determining the missing value as 6.

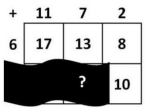
Compared to CoT and ToT, VReST demonstrates superior performance by leveraging multimodal fusion and systematic exploration. CoT struggles with intermediate verification, while ToT lacks effective feedback and global judgment. VReST addresses these shortcomings by incorporating MCTS, effectively integrating visual and textual information, and quantifying the reliability of reasoning traces. Across all cases, VReST not only achieves correct answers but also ensures interpretability and robustness, highlighting its effectiveness in solving complex vision-language reasoning tasks.

Algorithm 1 VReST

Require: Question Q, Image I, Max iterations K, Max depth D_{max}, Tree width w
Ensure: Final reasoning trace $\mathcal{P}^*$ and answer A^*

function VReST($Q, I, K, D_{\text{max}}, w$)
  Initialize search tree $\mathcal{T}$ with root node Q
  for $k = 1$ to K **do**
    $\mathcal{P}_{\text{selected}} \leftarrow$ Selection($\mathcal{T}$) ▷ UCT-based selection
    $\mathcal{P}_{\text{expanded}} \leftarrow$ Expansion($\mathcal{P}_{\text{selected}}, w, D_{\text{max}}$)
    $R \leftarrow$ SelfRewarding($\mathcal{P}_{\text{expanded}}, I$)
    Backpropagation($\mathcal{P}_{\text{expanded}}, R$)
  end for
  $\mathcal{P}^* \leftarrow$ FinalTraceSelection($\mathcal{T}$)
  return $\mathcal{P}^*, A^*$
end function
function SelfRewarding($\mathcal{P}, I$)
  $\mathcal{P}_t \leftarrow [Q, S_1, \ldots, S_t]$
  $R_1 \leftarrow P(\text{“Yes”}|[\mathcal{P}_t, \mathcal{P}_Q], I)$ ▷ Question usefulness
  $R_2 \leftarrow P(\text{“Yes”}|[\mathcal{P}_t, \mathcal{P}_A], I)$ ▷ Answer correctness
  return $\sqrt{R_1 R_2}$
end function
function Expansion($\mathcal{P}, w, D_{\text{max}}$)
  $\mathcal{P}_{\text{current}} \leftarrow \mathcal{P}$
  while not terminal and $|\mathcal{P}_{\text{current}}| < D_{\text{max}}$ **do**
    $\{S_{t,j}\}_{j=1}^{w} \leftarrow$ LVLM($\mathcal{P}_{\text{current}}, I$)
    $R_j \leftarrow$ SelfRewarding($[S_{t,j}], I$) for $j = 1, \ldots, w$
    $S_{t,\text{selected}} \leftarrow \arg\max_j R_j$
    $\mathcal{P}_{\text{current}} \leftarrow \mathcal{P}_{\text{current}} \cup \{S_{t,\text{selected}}\}$
  end while
  return $\mathcal{P}_{\text{current}}$
end function
function FinalTraceSelection($\mathcal{T}$)
  for each trace $\mathcal{P}$ in $\mathcal{T}$ **do**
    $R(\mathcal{P}) \leftarrow$ Avg($\{R(S_t)|S_t \in \mathcal{P}\}$)
  end for
  $\mathcal{P}^* \leftarrow \arg\max_{\mathcal{P}} R(\mathcal{P})$ ▷ Best Trace strategy
  $A^* \leftarrow$ extract answer from terminal node of $\mathcal{P}^*$
  return $\mathcal{P}^*, A^*$
end function

(a)



Question: Every box shows the result of the addition of the numbers on the very left and on the very top (for example: 6+2=8). Which number is written behind the question mark?

Ground Truth Answer： 15

CoT	**Sub-Q_1:** What is the pattern being applied in each box? **Sub-A_1**: The numbers in the boxes following this pattern are 17 (6+11), **8 (7+2), 21 (13+8)**, and ? (unknown). The answer is that the pattern is adding the bottom left and top right numbers.	**Sub-Q_2:** Which numbers from the pattern are relevant for solving the problem? **Sub-A_2**: The relevant numbers for solving the unknown box are **10 and 13**. The answer is 10 and 13.	**Sub-Q_3:** What is the sum of the relevant numbers? **Sub-A_3**: The sum of the relevant numbers 10 and 13 is 23. The answer is ***23***. ✗
ToT	**Sub-Q_1:** What is the rule for filling the boxes? **Sub-A_1**: The number in each box is the sum of the number from the top row and the number from the left column. The answer is: Box = Top Number + Left Number.	**Sub-Q_2:** Are the known boxes operations correct? **Sub-A_2**: Let's verify: 6 + 11 = 17, 6 + 7 = 13, 6 + 2 = 8. All of them are correct. The answer is' verified '.	**Sub-Q_3:** What is the value of the number written behind the question mark? **Sub-A_3**: **The number above the question** mark should be **4**, because 6+4=10. So 4+7=11. The final answer is ***11***. ✗
VReST	**Sub-Q_1:** What is the rule for filling the boxes? **Sub-A_1**: The number in each box is the sum of the number from the top row and the number from the left column. The answer is: Box = Top Number + Left Number.	**Sub-Q_2:** What is the missing number in the left column? **Sub-A_2**: From the bottom-right box, we know ?+2=10, so ?=10−2=8. The answer is: **The missing number is 8**.	**Sub-Q_3:** What number is in the missing box? **Sub-A_3**: Using the rule 8+7=15, the missing box contains 15.The answer is: ***15***. ✓

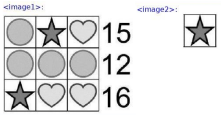
(b)

Question: How many different digits can you find in this picture?

Ground Truth Answer： 6

CoT	**Sub-Q_1:** What digits are visible in the drawing? **Sub-A_1**: The digits visible in the drawing are **0, 5, and 3**. The answer is 0, 5, and 3.	**Sub-Q_2:** How many unique digits are present in the drawing? **Sub-A_2**: The unique digits are 0, 5, and 3, which gives us 3 unique digits. The answer is 3.	**Sub-Q_3:** Now we can answer the question: How many different digits can you find in this picture? **Sub-A_3**: There are 3 different digits in the picture. The final answer is ***3***. ✗	
ToT	**Sub-Q_1:** What components of the drawing represent digits? **Sub-A_1**: The digits in the drawing appear in various parts, including the face **(0 and 5)** and the feet **(3)**. The answer is 0, 5, and 3.	**Sub-Q_2:** Are there any repeated digits in the drawing? **Sub-A_2**: Yes, the digit 0 is repeated in the eyes, but it is still counted as a single unique digit. The answer is yes.	**Sub-Q_3:** How many unique digits are present in total? **Sub-A_3**: After removing duplicates, the digits are 0, 5, and 3, which gives us 3 unique digits. The answer is ***3***. ✗	
VReST	**Sub-Q_1:** What digits can be identified in the drawing? **Sub-A_1**: The digits visible in the drawing are **0, 5, 3, and 2.** The answer is 0, 5, 3, and 2.	**Sub-Q_2:** Are there any other digits on the drawing? **Sub-A_2**: Yes, besides 0, 5, 3, 2, **there are also 8 and 9**. The answer is: The digits are 8 and 9.	**Sub-Q_3:** How many unique digits are present in total? **Sub-A_3**: The unique digits are 0, 5, 3, 2, 8 and 9. which makes 6 unique digits. **The answer is 6**.	**Sub-Q_4:** Now we can answer the question: How many different digits can you find in this picture? **Sub-A_4**: There are 6 different digits. The final answer is ***6***. ✓

(c)



Question: Every box shows the result of the addition of the numbers on the very left and on the very top (for example: $6+2=8$). Which number is written behind the question mark?

Ground Truth Answer： 6

CoT	**Sub-Q_1:** What does the second line tell us? **Sub-A_1**: **The sum of three circles is 12**, so **each circle represents 4**. The answer is 4.	**Sub-Q_2:** What is the first line equation? **Sub-A_2**: 4+Star+Heart=15, so **Star+Heart=11**. The answer is 11.	**Sub-Q_3:** Now we can answer the question: What number does the star represent? **Sub-A_3**: By solving the equation, we can conclude that the **star represents 7**. The final answer is ***7***. ✗	
ToT	**Sub-Q_1:** What equation can we write based on the rows? **Sub-A_1**: Circle + Star + Heart = 15, **2 × Circle + Star = 12**, Star + 2 × Heart = 16.The answer is the three equations above.	**Sub-Q_2:** What is the value of the Circle and Heart? **Sub-A_2**: From the second equation: 2 × Circle + Star = 12, and **substituting Circle = 3**, we find Heart = Circle + 3 = 6	**Sub-Q_3:** What is the value of the Star? **Sub-A_3**: Substitute Circle = 3 and Heart = 6 into the first equation: Circle + Star + Heart = 15, which becomes: 3 + Star + 6 = 15, so Star = 6.The answer is ***6***. ✓	
VReST	**Sub-Q_1:** What does the second line tell us? **Sub-A_1**: **The sum of three circles is 12**, so **each circle represents 4**. The answer is 4.	**Sub-Q_2:** What can we determine from row 1 using the circle value? **Sub-A_2**: Since circle=4, and 4 + s + h = 15, therefore **s + h = 11.** The answer is 11.	**Sub-Q_3:** What can we determine from row 3? **Sub-A_3**: Since s + h + h = 16, and we know s + h = 11, so **h = 5**. The answer is 5.	**Sub-Q_4:** Now we can answer the question: What number does the star represent? **Sub-A_4**: Since s + h = 11 and h = 5, then s = 6. Therefore, **the star represents 6.** The answer is ***6***. ✓

Figure 5: Case study comparing the reasoning results of CoT, ToT, and VReST frameworks. (a) involves determining the missing value in a grid based on the summation rule. (b) involves identifying all unique digits in a drawing based on visual patterns. (c) involves determining the missing value in a grid based on algebraic equations.

D.1 Prompt Template of Reasoning Step Generation

Instruction

Given a question, please decompose it into sub−questions. For each sub−question, please answer it in a complete sentence, ending with "The answer is". When the original question is answerable, please start the sub−question with "Now we can answer the question:".

∗∗Output Example:∗∗

∗∗Question:∗∗ Four years ago, Kody was only half as old as Mohamed. If Mohamed is currently twice as 30 years old, how old is Kody?

Sub−question 1: How old is Mohamed?
Answer 1: He is currently 30 ∗ 2 = 60 years old. The answer is 60.

Sub−question 2: How old was Mohamed four years ago?
Answer 2: Four years ago, he must have been 60 − 4 = 56 years old. The answer is 56.

Sub−question 3: How old is Kody four years ago?
Answer 3: Four years ago, Kody was half as old as Mohamed. So Kody was 56 / 2 = 28 years old. The answer is 28.

Sub−question 4: How old is Kody now?
Answer 4: Kody is 28 + 4 = 32 years old. The answer is 32.

Sub−question 5: Now we can answer the question: How old is Kody?
Answer 5: Kody is currently 32 years old. The final answer is 32.

Test example:

∗∗Question:∗∗ [question]

Sub−question 1: [sub−question 1]
Answer 1: [answer 1]
...
Sub−question k−1: [sub−question k−1]
Answer k−1: [answer k−1]

Answer:

Sub-question k: [sub-question k]
Answer k: [answer k]

D.2 Prompt Template of Calculating Usefulness of All the Sub-questions. (R1 Rewarding)

Instruction

Given a question and some sub−questions, determine whether the last sub−question is useful to answer the question. Output 'Yes' or 'No', and a reason.

∗∗Output Example:∗∗

∗∗Question:∗∗ Four years ago, Kody was only half as old as Mohamed. If Mohamed is currently twice as 30 years old, how old is Kody?
Sub−question 1: How old is Mohamed?
Sub−question 2: How old was Mohamed four years ago?
New Sub−question 3: How old was Kody four years ago?
Is the new question useful? Yes. We need the answer to calculate how old is Kody now.

∗∗Question:∗∗ Traci and Harris are baking cakes together. Traci has brought flour from her own house and Harris has 400g of flour in his house. Each cake needs 100g of flour and Traci and Harris have created 9 cakes each. How much flour, in grams, did Traci bring from her own house?
New Sub−question 1: How many cakes did Traci bring from her own house?
Is the new question useful? No. The new question is not related to the original question.

∗∗Question:∗∗ A quantity surveyor is figuring out the construction costs for a couple that wishes to build a house. The costs are as follows: land costs $50 per square meter, bricks cost $100 per 1000 bricks and roof tiles cost $10 per roof tile. If the house they wish to build requires 2000 square meters, 10000 bricks, and 500 roof tiles, how much construction costs are required for this project?
Sub−question 1: How much does the land cost?
Sub−question 2: How much do the bricks cost?
New Sub−question 3: How much do the roof tiles cost?
Is the new question useful? Yes. We need the answer to calculate the total construction costs.

∗∗Question:∗∗ Wallace's water heater is twice the size of Catherine's water heater. If the capacity of Wallace's water heater is 40 gallons and it's 3/4 full, calculate the total number of gallons of water they both have if Catherine's water heater is also full with water to 3/4 of its capacity.
Sub−question 1: How much water is in Wallace's water heater?
New Sub−question 2: How much water do they have in total?
Is the new question useful? No. It is too hard to answer the new question based on the current information.

Test example:

∗∗Question:∗∗ [question]
Sub−question 1: [sub−question 1]
Sub−question 2: [sub−question 2]
...
New Sub−question k: [sub−question k]
Is the new question useful?

Answer:

Yes/No. [reason]

D.3 Prompt Template of Calculating Correctness of the Last Answer. (R2 Rewarding)

Instruction

Given a question and some sub−questions and answers, determine whether the last answer of the last sub−question is correct. Output 'Yes' or 'No'.

Test example:

∗∗Question:∗∗ [question]
Sub−question 1: [sub−question 1]
Answer 1: [answer 1]
Sub−question 2: [sub−question 2]
Answer 2: [answer 2]
...
Sub−question k: [sub−question k]
Answer k: [answer k]
Is the answer correct?

Answer:

Yes/No.

D.4 Prompt Template for answer evaluation

Instruction

You will be given a ∗∗Question∗∗, the ∗∗Ground Truth Answer∗∗, and a ∗∗Predicted Answer∗∗. Your task is to compare the ∗∗Ground Truth Answer∗∗ with the ∗∗Predicted Answer∗∗ and determine whether the ∗∗Predicted Answer∗∗ is correct. It's acceptable to have different grammar or form. If the ∗∗Predicted Answer∗∗ is correct, you should say "Yes". If the ∗∗ Predicted Answer∗∗ is incorrect, you should say "No".

Test example:

∗∗Question:∗∗ [question]
∗∗Ground Truth Answer:∗∗ [ground_truth]
∗∗Predicted Answer:∗∗ [model_response]
Is the ∗∗Predicted Answer∗∗ correct?

Answer:

Yes/No.