

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 Reasoning in LVLMs through Monte Carlo Tree Search and Self-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,

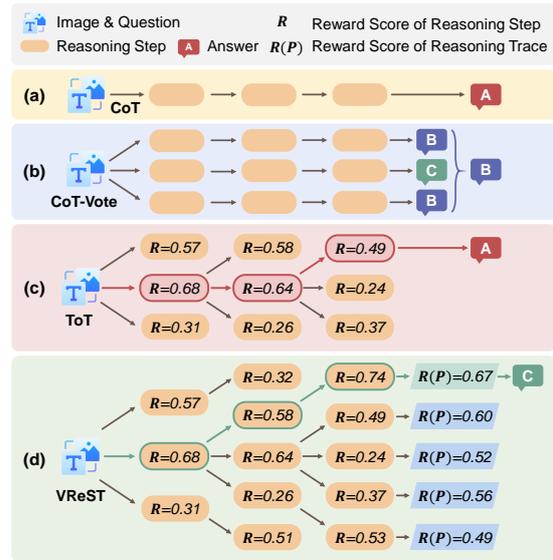


Figure 1: The difference between VReST and the previous multimodal CoT prompting methods. The methods in (a)(b)(c) obtain suboptimal solutions by a greedy algorithm, while VReST in (d) can fully explore the reasoning space to obtain the optimal solution.

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.

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

062	Carlo Tree Search (MCTS) (Hao et al., 2023;	uses MCTS to enhance the depth and qual-	114
063	Zhang et al., 2024b; Jiang et al., 2024; Long, 2023;	ity of reasoning in LVLMs.	115
064	Yao et al., 2024) can effectively expand the reason-	• We propose a Self-Reward mechanism incor-	116
065	ing space in a training-free manner, improving CoT	porating visual information to evaluate reason-	117
066	generation. Based on these findings, we extend	ing traces.	118
067	the MCTS algorithm to LVLm. A key component	• We achieve SOTA performance on three mul-	119
068	of any tree search algorithm is the reward func-	timodal mathematical reasoning datasets, out-	120
069	tion, which guides the model’s exploration within	performing existing prompting methods.	121
070	the vast space of possible reasoning traces (Feng	• We demonstrate that VReST exhibits a better	122
071	et al., 2023). To ensure a fair comparison with	test-time scaling law in multimodal tasks.	123
072	baseline methods, we avoid introducing additional		
073	models. Hence, we propose a multimodal Self-	2 Related Work	124
074	Reward mechanism that incorporates visual knowl-		
075	edge with textual clues.	2.1 CoT for Large Vision-Language Models	125
076	To tackle the intricacies of complex vision tasks	Large Vision-Language Models (LVLms) demon-	126
077	within LVLms, we introduce VReST , a pioneer-	strate remarkable abilities in integrating visual and	127
078	ing approach that Enhancing Reasoning in Large	linguistic information (Li et al., 2024; Peng et al.,	128
079	Vision-Language Models through Tree Search and	2024), but face challenges in tasks requiring com-	129
080	Self-Reward mechanism. Figure 1 shows the dif-	plex reasoning or multi-hop inferences (Lu et al.,	130
081	ference between VReST and existing multimodal	2023; Wang et al., 2024a,c; Zhao et al., 2024; Chen	131
082	CoT methods. VReST employs MCTS to system-	et al., 2024). Extending the Chain of Thought	132
083	atically navigate the reasoning space, where nodes	(CoT) paradigm (Kojima et al., 2022; Zhang et al.,	133
084	symbolize individual reasoning steps, and paths	2022) to the multimodal domain offers a promis-	134
085	constitute complete reasoning trajectories. By re-	ing direction. While many approaches enhance the	135
086	curisively identifying nodes with high confidence,	CoT reasoning abilities of LVLms through exten-	136
087	VReST dynamically crafts reasoning steps and fos-	sive training (Xu et al., 2023; Shao et al., 2024;	137
088	ters diversity by modulating the temperature of	Cheng et al., 2024; Guo et al., 2024), optimiz-	138
089	LVLm generation, thus enriching the exploration	ing reasoning traces provides a viable training-free	139
090	of the reasoning space. Based on prior work (Hao	alternative. Initial effort adopts a two-stage reason-	140
091	et al., 2023), we present a multimodal Self-Reward	ing method (Zhang et al., 2023) where rationales	141
092	mechanism that appraises the merit of reasoning	precede the final answer to enable step-by-step in-	142
093	steps. It considers sub-question utility, final answer	ference. Subsequent advancements augment reason-	143
094	correctness, and vision-language clues. Inspired	ing steps with precise visual details, such as	144
095	by (Lightman et al., 2023), our mechanism assigns	scene graphs (Mitra et al., 2024) and related image	145
096	reward values to each node.	regions (Shao et al., 2024). To better understand	146
097	Finally, VReST expands, evaluates, and back-	textual information, DDCoT (Zheng et al., 2023)	147
098	propagates reasoning traces in each iteration,	decomposes questions into sub-questions, and uti-	148
099	thereby refining the search tree by updating node	lize sub-answers to construct reasoning steps. Can-	149
100	statistics. The optimal reasoning trace is selected	tor (Gao et al., 2024) further improves this ap-	150
101	based on the aggregate reward, with the final an-	proach by framing LVLms as multifaceted experts	151
102	swer being extracted from the terminal node. Ex-	for multi-step reasoning.	152
103	periments show that VReST outperforms exist-	However, these methods struggle with complex	153
104	ing prompting methods on three visual reasoning	questions due to limited reasoning steps and lack of	154
105	datasets. Moreover, as shown in Section 4.7, the	feedback to refine traces. VReST addresses these	155
106	performance gain of our approach becomes more	issues with a tree search for extended reasoning	156
107	pronounced with increasing iterations of MCTS,	and reward evaluation for optimal solutions.	157
108	surpassing other prompting methods, and demon-		
109	strating better multimodal test-time scaling. Our	2.2 Tree-based Reasoning with LLMs	158
110	approach offers a promising direction for training-	Tree-based reasoning methods enhance perfor-	159
111	free methods to enhance LVLm reasoning.	mance by increasing computational costs to explore	160
112	Our main contributions are as follows:	diverse solution spaces (Jiang et al., 2024). Self-	161
113	• We introduce a training-free approach that	Consistency (Wang et al., 2022) improves accuracy	162

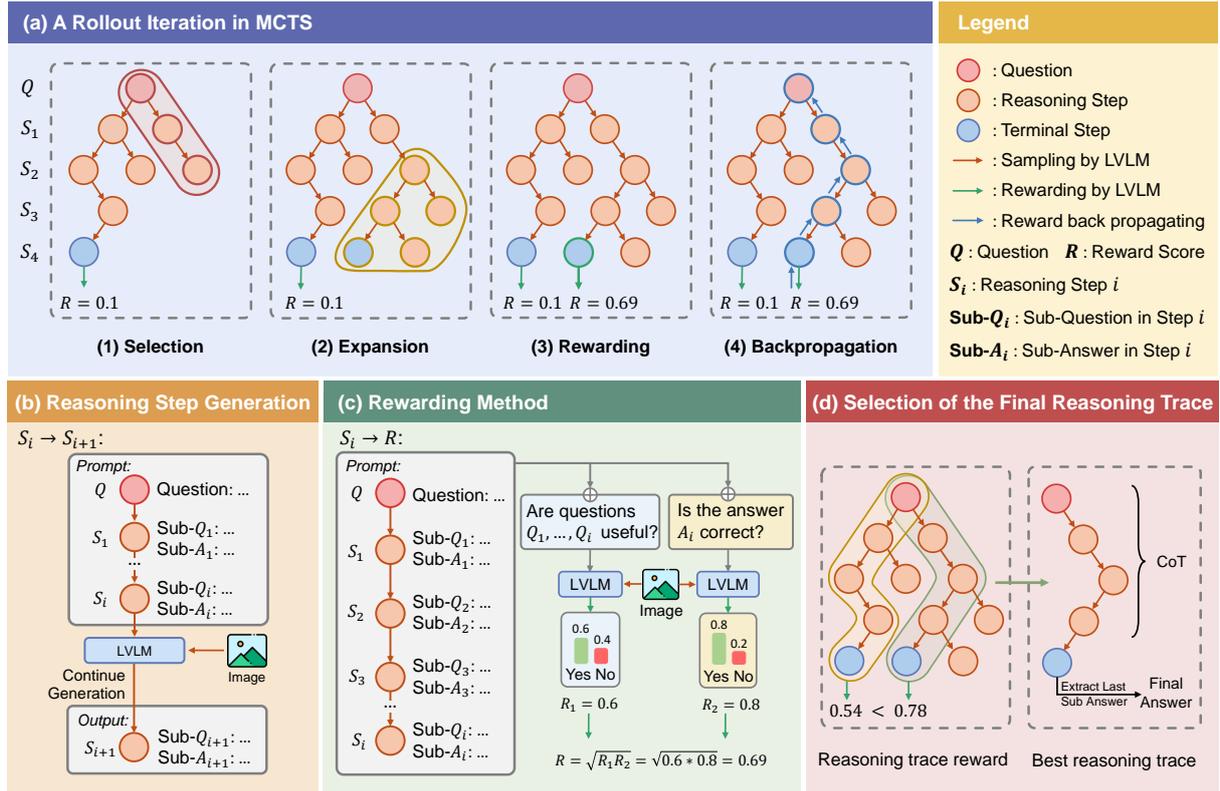


Figure 2: The framework of VReST. (a) Illustrates the MCTS rollout iteration process, including Selection, Expansion, Rewarding, and Backpropagation steps. (b) Depicts the generation of new reasoning steps using LVLm based on the constructed prompt. (c) Shows the Self-Rewarding mechanism for calculating 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.

163 by sampling multiple reasoning traces, while Tree
 164 of Thoughts (ToT) (Long, 2023; Yao et al., 2024)
 165 use heuristic methods to select optimal steps but of-
 166 ten converges to locally optimal solutions. Breadth-
 167 First Search(BFS) (Yao et al., 2024) identifies glob-
 168 ally optimal reasoning traces by exploring the en-
 169 tire space. Monte Carlo Tree Search (MCTS) (Hao
 170 et al., 2023; Zhang et al., 2024b) further integrates
 171 rewarding and backpropagation mechanisms, quan-
 172 tifying each inference trace across multiple iter-
 173 ations to identify the globally optimal solution. De-
 174 spite their potential, tree-based reasoning methods
 175 have rarely been applied to multimodal reasoning
 176 tasks. Our framework incorporates visual inform-
 177 ation into reasoning steps and, to the best of our
 178 knowledge, is the first to employ MCTS for multi-
 179 modal CoT reasoning.

3 Method

181 As shown in Figure 2, our approach combines
 182 Monte Carlo Tree Search (MCTS) with Large
 183 Vision-Language Model (LVLm) to generate step-
 184 by-step reasoning traces and evaluate them using a

185 Self-Rewarding mechanism. Below, we detail the
 186 problem formulation (3.1), the MCTS framework
 187 with a Self-Reward mechanism (3.2), as well as the
 188 final reasoning trace selection method (3.3).

3.1 Problem Formulation

189 Given a question Q and an image I , our goal is to
 190 find the optimal reasoning trace \mathcal{P}^* that leads to the
 191 correct answer A . Each reasoning trace \mathcal{P} consists
 192 of an original question and a sequence of reason-
 193 ing steps: $\{Q, S_1, S_2, \dots, S_n\}$, where each step S_i
 194 contains a sub-question Q_i and its corresponding
 195 sub-answer A_i .
 196

3.2 Monte Carlo Tree Search Framework

197 In Figure 2(a), we employ MCTS to explore the
 198 reasoning space systematically. Each node in the
 199 search tree represents a reasoning step S_i , and
 200 edges represent the transitions between steps. The
 201 rollout iteration in MCTS involves four steps: Se-
 202 lection, Expansion, Rewarding, and Backpropaga-
 203 tion. These steps are iteratively performed K times
 204 to explore the reasoning space and refine the search
 205

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:

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

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:

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

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 G.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:

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

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, \dots, 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 $t = 1, \dots, T$, we update its statistics by aggregating the rewards in all future steps of S_t :

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

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	<u>51.39</u>	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	<u>43.06</u>	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	<u>54.39</u>	13.51	43.75	57.38	74.09	60.20
VReST	68.03	56.73	<u>72.04</u>	67.09	<u>58.10</u>	59.43	<u>62.61</u>	58.16	29.73	50.69	<u>67.21</u>	75.75	<u>64.50</u>
VReST-Vote	69.14	<u>51.44</u>	75.81	<u>66.46</u>	64.25	<u>54.45</u>	67.42	53.56	<u>27.03</u>	60.42	68.03	<u>77.74</u>	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:

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

And then select the trace with the highest value:

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

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), **Math-Vision** (Wang et al., 2024a) and **CharXiv** (Wang et al., 2024c). All datasets are evaluated using answer accuracy. See Appendix B 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 C 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 G.

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	<u>15.79</u>	15.79	10.53	<u>21.05</u>	0.00	5.26	5.26	21.05	15.79	57.89	15.79	36.84	15.79	15.79	15.79	<u>26.32</u>	18.42
CoT	<u>15.79</u>	10.53	<u>15.79</u>	10.53	15.79	10.53	<u>26.32</u>	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	<u>31.58</u>	10.53	<u>21.05</u>	<u>31.58</u>	31.58	21.05	21.71
Best-of-N	5.26	<u>31.58</u>	0.00	<u>21.05</u>	<u>21.05</u>	26.32	<u>26.32</u>	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	<u>15.79</u>	<u>21.05</u>	<u>21.05</u>	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	<u>31.58</u>	21.05	<u>21.05</u>	15.79	10.53	10.53	42.11	42.11	15.79	36.84	10.53	26.32	<u>31.58</u>	52.63	36.84	<u>26.64</u>
VReST-Vote	10.53	42.11	<u>15.79</u>	31.58	<u>21.05</u>	<u>21.05</u>	36.84	<u>36.84</u>	<u>26.32</u>	<u>42.11</u>	26.32	<u>31.58</u>	15.79	<u>31.58</u>	<u>36.84</u>	<u>26.32</u>	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	<u>33.33</u>	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	<u>33.63</u>	27.30
CoT-Vote	32.95	45.45	28.88	22.71	26.98	28.99	33.61	30.37	39.37	<u>29.37</u>	25.86	32.74	30.90
Best-of-N	<u>34.09</u>	48.48	28.02	<u>24.02</u>	<u>33.33</u>	30.43	30.25	35.56	<u>38.58</u>	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	<u>30.17</u>	27.43	29.00
ToT	<u>34.09</u>	45.45	<u>33.62</u>	20.96	30.95	26.81	36.97	31.85	35.43	29.37	26.72	39.82	32.10
VReST	33.64	<u>54.55</u>	<u>33.62</u>	22.27	30.95	<u>31.16</u>	<u>41.18</u>	<u>40.74</u>	33.86	26.98	<u>30.17</u>	29.20	<u>33.10</u>
VReST-Vote	37.95	61.62	39.22	27.07	37.30	38.41	45.38	43.70	<u>38.58</u>	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.

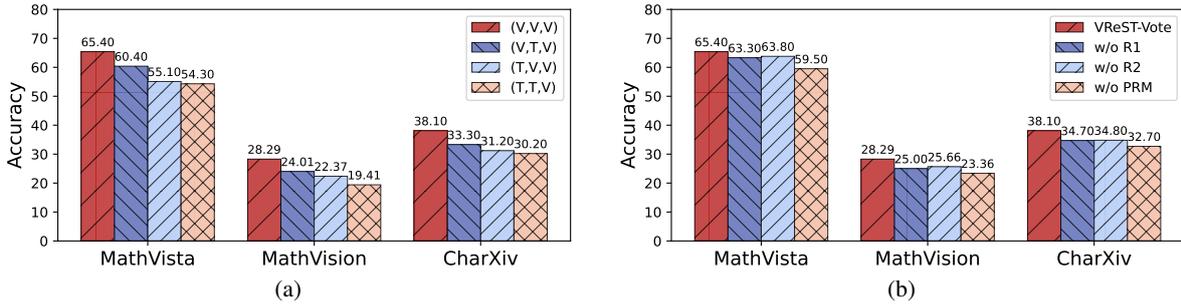


Figure 3: (a) Ablation results of different configurations of visual and text-only components. (V, V, V) represents using LVLm for all components (reasoning generation, R1, and R2 rewarding), while T denotes using text-only LLM. (b) Ablation results from different reward methods. w/o R1 and w/o R2 denote 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

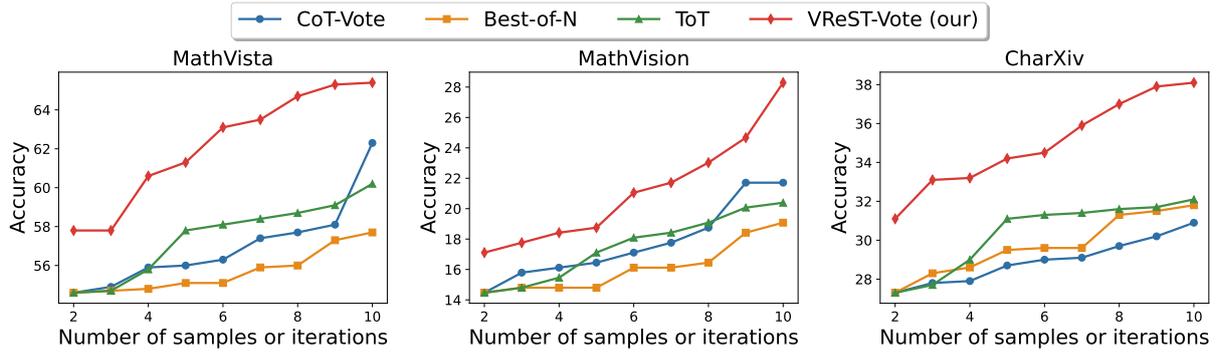


Figure 4: The impact of the number of samples or iterations. It shows that our VReST exhibits a better test-time scaling law than other SOTA methods in multimodal reasoning tasks.

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.

549 Limitations

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

553 **Self-Reward Mechanism** To ensure a fair compar-
554 ison with baseline methods, we designed the
555 self-reward mechanism to use the LVLM itself
556 for reward scoring, without introducing additional
557 models. This approach aligns with the training-
558 free nature of our method, enabling quick deploy-
559 ment without the need for training a separate re-
560 ward model. However, this mechanism heavily
561 relies on the LVLM’s own judgments to evaluate
562 the quality of reasoning traces. As a result, there is
563 a risk that model biases or errors could propagate
564 through the reward process, potentially affecting
565 the accuracy and reliability of the reasoning pro-
566 cess. Future work could involve training an addi-
567 tional reward model to assist the LVLM’s reasoning
568 process, helping to mitigate potential biases and
569 improve the accuracy of the reward signal.

570 **Computational Cost** The MCTS approach re-
571 lies on multiple iterations and extensive tree explo-
572 ration, resulting in significant computational over-
573 head compared to current prompting methods. This
574 increased cost may limit the scalability of VReST
575 for large-scale applications. In future work, we
576 aim to address this by incorporating pruning strate-
577 gies or early stopping techniques within the tree
578 search process, which could help reduce the com-
579 putational burden while maintaining performance.

580 **Model Dependency** Currently, we have only
581 evaluated the effectiveness of VReST on the
582 Qwen2-VL-7B-Instruct model. Although this
583 model demonstrates the benefits of our approach,
584 the effectiveness of VReST may vary across dif-
585 ferent LVLMs, especially models with different
586 architectures, scales, or training regimens. In fu-
587 ture work, further experimentation on a wider range
588 of LVLMs will be essential to determine the gener-
589 alizability of our approach.

590 **Dataset Dependency** Our experiments primarily
591 focus on a limited set of visual reasoning datasets.
592 While VReST shows promising results on these
593 datasets, its performance on other datasets with
594 different characteristics, such as those involving
595 diverse types of reasoning or tasks outside visual
596 reasoning, remains unexplored. Expanding our
597 evaluation to a broader set of datasets will help as-

sess 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. M³CoT: 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, Wenhui 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.

650	Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri	chain of thought reasoning in language models . In	705
651	Edwards, Bowen Baker, Teddy Lee, Jan Leike,	The Eleventh International Conference on Learning	706
652	John Schulman, Ilya Sutskever, and Karl Cobbe.	Representations .	707
653	2023. Let's verify step by step. arXiv preprint		
654	arXiv:2305.20050 .		
655	Mengsha Liu, Daoyuan Chen, Yaliang Li, Guian Fang,	Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen,	708
656	and Ying Shen. 2024. Chartthinker: A contextual	Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu,	709
657	chain-of-thought approach to optimized chart sum-	Haotian Liu, Sadhika Malladi, et al. 2024c. Charxiv:	710
658	marization. arXiv preprint arXiv:2403.11236 .	Charting gaps in realistic chart understanding in mul-	711
		timodal llms. arXiv preprint arXiv:2406.18521 .	712
659	Jieyi Long. 2023. Large language model guided tree-of-	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	713
660	thought. arXiv preprint arXiv:2305.08291 .	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,	714
		et al. 2022. Chain-of-thought prompting elicits	715
661	Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chun-	reasoning in large language models. Advances in	716
662	yuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-	Neural Information Processing Systems , 35:24824–	717
663	Wei Chang, Michel Galley, and Jianfeng Gao. 2023.	24837.	718
664	Mathvista: Evaluating mathematical reasoning of	Yifan Wu, Lutao Yan, Yuyu Luo, Yunhai Wang, and Nan	719
665	foundation models in visual contexts. arXiv preprint	Tang. 2024. Evaluating task-based effectiveness of	720
666	arXiv:2310.02255 .	mllms on charts. arXiv preprint arXiv:2405.07001 .	721
667	Chancharik Mitra, Brandon Huang, Trevor Darrell,	Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun	722
668	and Roei Herzig. 2024. Compositional chain-of-	Yuan, and Jian Guo. 2023. Chartbench: A bench-	723
669	thought prompting for large multimodal models.	mark for complex visual reasoning in charts. arXiv	724
670	In Proceedings of the IEEE/CVF Conference on	preprint arXiv:2312.15915 .	725
671	Computer Vision and Pattern Recognition , pages		
672	14420–14431.	An Yang, Baosong Yang, Binyuan Hui, Bo Zheng,	726
673	OpenAI. 2024. Introducing openai o1-preview . Ac-	Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan	727
674	cessed: 2024-12-13.	Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2	728
675	Yingzhe Peng, Xinting Hu, Jiawei Peng, Xin Geng,	technical report. arXiv preprint arXiv:2407.10671 .	729
676	Xu Yang, et al. 2024. Live: Learnable in-	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran,	730
677	context vector for visual question answering. In	Tom Griffiths, Yuan Cao, and Karthik Narasimhan.	731
678	The Thirty-eighth Annual Conference on Neural	2024. Tree of thoughts: Deliberate problem solving	732
679	Information Processing Systems .	with large language models. Advances in Neural	733
680	Hao Shao, Shengju Qian, Han Xiao, Guanglu	Information Processing Systems , 36.	734
681	Song, Zhuofan Zong, Letian Wang, Yu Liu, and	Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue,	735
682	Hongsheng Li. 2024. Visual cot: Advancing	Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm	736
683	multi-modal language models with a comprehen-	self-training via process reward guided tree search.	737
684	sive dataset and benchmark for chain-of-thought	arXiv preprint arXiv:2406.03816 .	738
685	reasoning. In The Thirty-eight Conference on	Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jia-	739
686	Neural Information Processing Systems Datasets	tong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang,	740
687	and Benchmarks Track .	Marco Pavone, Yuqiang Li, et al. 2024b. Llama-	741
688	Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie	berry: Pairwise optimization for o1-like olympiad-	742
689	Zhan, and Hongsheng Li. 2024a. Measuring mul-	level mathematical reasoning. arXiv preprint	743
690	timodal mathematical reasoning with math-vision	arXiv:2410.02884 .	744
691	dataset. arXiv preprint arXiv:2402.14804 .	Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun	745
692	Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhi-	Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu,	746
693	hao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin	Kai-Wei Chang, Yu Qiao, et al. 2025. Mathverse:	747
694	Wang, Wenbin Ge, et al. 2024b. Qwen2-vl: Enhanc-	Does your multi-modal llm truly see the diagrams in	748
695	ing vision-language model's perception of the world	visual math problems? In European Conference on	749
696	at any resolution. arXiv preprint arXiv:2409.12191 .	Computer Vision , pages 169–186. Springer.	750
697	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le,	Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex	751
698	Ed Chi, Sharan Narang, Aakanksha Chowdhery, and	Smola. 2022. Automatic chain of thought prompt-	752
699	Denny Zhou. 2022. Self-consistency improves chain	ing in large language models. arXiv preprint	753
700	of thought reasoning in language models. arXiv	arXiv:2210.03493 .	754
701	preprint arXiv:2203.11171 .	Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao,	755
702	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le,	George Karypis, and Alex Smola. 2023. Multi-	756
703	Ed H. Chi, Sharan Narang, Aakanksha Chowdhery,	modal chain-of-thought reasoning in language mod-	757
704	and Denny Zhou. 2023. Self-consistency improves	els. arXiv preprint arXiv:2302.00923 .	758

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

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

A Discussion

In this section, we will address the following Discussion Questions (DQ) to elucidate our contributions more clearly.

DQ1: Why do we not need any additional baselines?

See Appendix C, where we list all the baselines used to compare with our proposed method. However, we did not compare with other methods using MCTS.

At present, many MCTS-based methods require training the LLM itself or a reward model, such as ReST-MCTS* (Zhang et al., 2024a) and LLaMA-Berry (Zhang et al., 2024b), among others. Our work, as an initial endeavor to introduce MCTS into multimodal reasoning tasks, primarily focuses on training-free methods. To ensure a fair comparison, we have opted not to incorporate additional reward models, thereby maintaining the comparability of our experiments with baseline methods.

Consequently, under the training-free setting, we have taken into account as many methods as possible, and we believe that the current baselines are sufficiently comprehensive.

DQ2: Why the datasets we have chosen are sufficient to demonstrate the effectiveness of VReST?

See Appendix B, where we list all the datasets used in this paper. While our primary evaluation focuses on mathematical reasoning, the datasets we selected (MathVista, MathVision, and CharXiv) actually cover a broad spectrum of multimodal reasoning tasks. For example:

MathVista includes various task types like figure question answering (FQA), geometry problem solving (GPS), and visual question answering (VQA).

CharXiv contains diverse chart understanding tasks involving both descriptive and complex reasoning questions.

MathVision covers 16 distinct mathematical disciplines including topology, graph theory, and geometric reasoning.

Therefore, these three datasets can largely prove the effectiveness of our method.

B 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 mathe-

tical 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.

C 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 G.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 G.1. The width of the tree in ToT is $w = 10$, and the maximum depth in ToT is $D_{max} = 8$.

D Algorithm

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

Methods	MathVista	MathVision	CharXiv
QA	1.44	2.46	1.84
CoT	7.28	10.57	9.04
CoT-Vote	15.32	24.18	19.36
Best-of-N	18.56	28.50	21.07
Cantor	21.46	36.39	32.67
ToT	34.39	45.89	39.29
VReST	108.87	157.67	127.58

Table 5: The average time (in seconds) for the different methods to complete a sample.

E Time efficiency analysis

Table 5 shows the average time (in seconds) for the different methods to complete a sample. Note that VReST and VReST-Vote only differ in the Final Reasoning Trace Selection stage, so both take the same time to complete a sample.

While VReST does require more computation than standard CoT approaches, we believe the performance gains justify the additional cost.

F More Experimental Results

As shown in Tables 6, 7, 8, we also provide experimental results on the smaller model Qwen2.5-VL-3B-Instruct. Experimental results show that our method is still effective on smaller scale models.

G 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 G.1. For the prompts in Appendix G.1, G.2, and G.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 G.4 to judge whether the final answer is correct or not.

G.1 Reasoning Step Generation

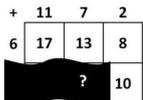
As shown in the Prompt Template of Reasoning Step Generation, we input $k - 1$ sub-questions and corresponding answers and let LVLm continue to generate the k -th sub-question and corresponding answer. Model-generated content is annotated in blue.

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^*

```
1: function VREST( $Q, I, K, D_{\max}, w$ )
2:   Initialize search tree  $\mathcal{T}$  with root node  $Q$ 
3:   for  $k = 1$  to  $K$  do
4:      $\mathcal{P}_{\text{selected}} \leftarrow \text{SELECTION}(\mathcal{T})$  ▷ UCT-based selection
5:      $\mathcal{P}_{\text{expanded}} \leftarrow \text{EXPANSION}(\mathcal{P}_{\text{selected}}, w, D_{\max})$ 
6:      $R \leftarrow \text{SELFREWARDING}(\mathcal{P}_{\text{expanded}}, I)$ 
7:      $\text{BACKPROPAGATION}(\mathcal{P}_{\text{expanded}}, R)$ 
8:   end for
9:    $\mathcal{P}^* \leftarrow \text{FINALTRACESELECTION}(\mathcal{T})$ 
10:  return  $\mathcal{P}^*, A^*$ 
11: end function
12: function SELFREWARDING( $\mathcal{P}, I$ )
13:   $\mathcal{P}_t \leftarrow [Q, S_1, \dots, S_t]$ 
14:   $R_1 \leftarrow P(\text{"Yes"} | [\mathcal{P}_t, \mathcal{P}_Q], I)$  ▷ Question usefulness
15:   $R_2 \leftarrow P(\text{"Yes"} | [\mathcal{P}_t, \mathcal{P}_A], I)$  ▷ Answer correctness
16:  return  $\sqrt{R_1 R_2}$ 
17: end function
18: function EXPANSION( $\mathcal{P}, w, D_{\max}$ )
19:   $\mathcal{P}_{\text{current}} \leftarrow \mathcal{P}$ 
20:  while not terminal and  $|\mathcal{P}_{\text{current}}| < D_{\max}$  do
21:     $\{S_{t,j}\}_{j=1}^w \leftarrow \text{LVLM}(\mathcal{P}_{\text{current}}, I)$ 
22:     $R_j \leftarrow \text{SELFREWARDING}([S_{t,j}], I)$  for  $j = 1, \dots, w$ 
23:     $S_{t,\text{selected}} \leftarrow \arg \max_j R_j$ 
24:     $\mathcal{P}_{\text{current}} \leftarrow \mathcal{P}_{\text{current}} \cup \{S_{t,\text{selected}}\}$ 
25:  end while
26:  return  $\mathcal{P}_{\text{current}}$ 
27: end function
28: function FINALTRACESELECTION( $\mathcal{T}$ )
29:  for each trace  $\mathcal{P}$  in  $\mathcal{T}$  do
30:     $R(\mathcal{P}) \leftarrow \text{AVG}(\{R(S_t) | S_t \in \mathcal{P}\})$ 
31:  end for
32:   $\mathcal{P}^* \leftarrow \arg \max_{\mathcal{P}} R(\mathcal{P})$  ▷ Best Trace strategy
33:   $A^* \leftarrow \text{extract answer from terminal node of } \mathcal{P}^*$ 
34:  return  $\mathcal{P}^*, A^*$ 
35: 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-Q₁: What is the pattern being applied in each box?
Sub-A₁: 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-Q₂: Which numbers from the pattern are relevant for solving the problem?
Sub-A₂: The relevant numbers for solving the unknown box are 10 and 13. The answer is 10 and 13.

Sub-Q₃: What is the sum of the relevant numbers?
Sub-A₃: The sum of the relevant numbers 10 and 13 is 23. The answer is 23. ❌

ToT

Sub-Q₁: What is the rule for filling the boxes?
Sub-A₁: 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-Q₂: Are the known boxes operations correct?
Sub-A₂: Let's verify: $6 + 11 = 17$, $6 + 7 = 13$, $6 + 2 = 8$. All of them are correct. The answer is 'verified'.

Sub-Q₃: What is the value of the number written behind the question mark?
Sub-A₃: The number above the question mark should be 4, because $6+4=10$. So $4+7=11$. The final answer is 11. ❌

VReST

Sub-Q₁: What is the rule for filling the boxes?
Sub-A₁: 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-Q₂: What is the missing number in the left column?
Sub-A₂: From the bottom-right box, we know $?+2=10$, so $?=10-2=8$. The answer is: The missing number is 8.

Sub-Q₃: What number is in the missing box?
Sub-A₃: 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-Q₁: What digits are visible in the drawing?
Sub-A₁: The digits visible in the drawing are 0, 5, and 3. The answer is 0, 5, and 3.

Sub-Q₂: How many unique digits are present in the drawing?
Sub-A₂: The unique digits are 0, 5, and 3, which gives us 3 unique digits. The answer is 3.

Sub-Q₃: Now we can answer the question: How many different digits can you find in this picture?
Sub-A₃: There are 3 different digits in the picture. The final answer is 3. ❌

ToT

Sub-Q₁: What components of the drawing represent digits?
Sub-A₁: 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-Q₂: Are there any repeated digits in the drawing?
Sub-A₂: Yes, the digit 0 is repeated in the eyes, but it is still counted as a single unique digit. The answer is yes.

Sub-Q₃: How many unique digits are present in total?
Sub-A₃: After removing duplicates, the digits are 0, 5, and 3, which gives us 3 unique digits. The answer is 3. ❌

VReST

Sub-Q₁: What digits can be identified in the drawing?
Sub-A₁: The digits visible in the drawing are 0, 5, 3, and 2. The answer is 0, 5, 3, and 2.

Sub-Q₂: Are there any other digits on the drawing?
Sub-A₂: Yes, besides 0, 5, 3, 2, there are also 8 and 9. The answer is: The digits are 8 and 9.

Sub-Q₃: How many unique digits are present in total?
Sub-A₃: The unique digits are 0, 5, 3, 2, 8 and 9, which makes 6 unique digits. The answer is 6.

Sub-Q₄: Now we can answer the question: How many different digits can you find in this picture?
Sub-A₄: 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-Q₁: What does the second line tell us?
Sub-A₁: The sum of three circles is 12, so each circle represents 4. The answer is 4.

Sub-Q₂: What is the first line equation?
Sub-A₂: $4+Star+Heart=15$, so $Star+Heart=11$. The answer is 11.

Sub-Q₃: Now we can answer the question: What number does the star represent?
Sub-A₃: By solving the equation, we can conclude that the star represents 7. The final answer is 7. ❌

ToT

Sub-Q₁: What equation can we write based on the rows?
Sub-A₁: Circle + Star + Heart = 15, $2 \times Circle + Star = 12$, $Star + 2 \times Heart = 16$. The answer is the three equations above.

Sub-Q₂: What is the value of the Circle and Heart?
Sub-A₂: From the second equation: $2 \times Circle + Star = 12$, and substituting Circle = 3, we find Heart = Circle + 3 = 6

Sub-Q₃: What is the value of the Star?
Sub-A₃: 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-Q₁: What does the second line tell us?
Sub-A₁: The sum of three circles is 12, so each circle represents 4. The answer is 4.

Sub-Q₂: What can we determine from row 1 using the circle value?
Sub-A₂: Since circle=4, and $4 + s + h = 15$, therefore $s + h = 11$. The answer is 11.

Sub-Q₃: What can we determine from row 3?
Sub-A₃: Since $s + h + h = 16$, and we know $s + h = 11$, so $h = 5$. The answer is 5.

Sub-Q₄: Now we can answer the question: What number does the star represent?
Sub-A₄: 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.

Methods	FQA	GPS	MWP	TQA	VQA	ALG	ARI	GEO	LOG	NUM	SCI	STA	ALL
QA	66.91	64.42	58.60	53.16	51.40	56.23	52.12	61.92	21.62	39.58	62.30	71.10	59.90
CoT	65.43	63.46	56.99	51.90	50.84	55.52	50.99	60.67	18.92	39.58	60.66	69.44	58.70
CoT-Vote	69.89	67.79	65.05	58.86	55.31	60.50	57.51	65.27	29.73	43.75	68.85	74.42	64.20
Best-of-N	69.52	66.35	60.75	55.06	53.07	58.01	54.39	65.27	29.73	41.67	64.75	72.43	62.00
Cantor	69.14	68.75	61.29	58.86	54.19	61.21	55.24	66.53	24.32	43.75	66.39	73.75	63.30
ToT	69.14	70.19	67.74	60.13	57.54	63.35	58.64	67.36	24.32	47.22	68.03	74.42	65.60
VReST	74.35	71.15	67.20	58.86	55.31	64.41	59.21	67.78	37.84	45.14	66.39	77.08	66.50
VReST-Vote	72.12	72.12	67.20	63.92	58.10	65.48	60.34	69.04	40.54	48.61	72.13	74.75	67.40

Table 6: Accuracy scores (%) on the testmini subset of MathVista on the Qwen2.5-VL-3B-Instruct.

Methods	ALG	AnaG	Ari	CombG	Comb	Cnt	DescG	GrphT	Log	Angle	Area	Len	SolG	Stat	Topo	TransG	ALL
QA	26.32	10.53	21.05	26.32	5.26	10.53	31.58	10.53	26.32	42.11	15.79	26.32	15.79	21.05	10.53	26.32	20.39
CoT	26.32	10.53	15.79	26.32	0.00	5.26	26.32	10.53	26.32	31.58	10.53	21.05	15.79	15.79	5.26	26.32	17.11
CoT-Vote	26.32	15.79	26.32	31.58	10.53	10.53	31.58	10.53	26.32	57.89	15.79	26.32	15.79	26.32	10.53	26.32	23.03
Best-of-N	31.58	10.53	21.05	26.32	5.26	10.53	36.84	10.53	26.32	42.11	15.79	26.32	15.79	26.32	10.53	26.32	21.38
Cantor	21.05	5.26	21.05	15.79	5.26	10.53	26.32	5.26	15.79	31.58	15.79	15.79	5.26	10.53	10.53	26.32	15.13
ToT	31.58	21.05	26.32	26.32	10.53	10.53	31.58	26.32	26.32	42.11	21.05	26.32	15.79	26.32	10.53	26.32	23.68
VReST	42.11	15.79	21.05	31.58	15.79	15.79	31.58	26.32	31.58	57.89	15.79	31.58	31.58	26.32	15.79	36.84	27.96
VReST-Vote	31.58	15.79	31.58	31.58	15.79	31.58	47.37	26.32	26.32	57.89	26.32	36.84	26.32	31.58	15.79	26.32	29.93

Table 7: Accuracy scores (%) on the testmini subset of MATH-Vision on the Qwen2.5-VL-3B-Instruct.

G.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 LVLMM and let it judge whether the new sub-question is useful or not. Model-generated content is annotated in blue.

G.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 LVLMM and let it judge whether the last answer is correct or not. Model-generated content is annotated in blue.

G.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.

H 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.

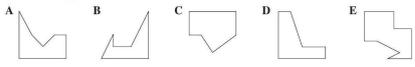
In Case 3, which involves solving a grid of algebraic equations, CoT’s linear reasoning misses critical steps, leading to an incorrect answer of 7. ToT applies tree-based reasoning but inadequately propagates constraints, yielding 11 as the result. VReST, however, integrates equations like $4+7+?=11$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

We demonstrate a bad case from Figure 6 where, despite the final output from VReST being incorrect, the root cause of the error lies in the in-

Methods	Text in Chart	Text in General	Num in Chart	Num in General	CS	EC	EESS	MATH	PHY	QB	QF	STA	ALL
QA	30.45	47.47	31.90	21.83	30.95	32.61	27.73	31.11	34.65	30.95	27.59	27.43	30.50
CoT	28.18	44.44	29.31	20.96	28.57	29.71	25.21	28.89	33.86	27.78	26.72	25.66	28.40
CoT-Vote	31.82	49.49	33.19	24.02	33.33	34.06	30.25	31.85	36.22	32.54	27.59	30.09	32.10
Best-of-N	32.73	48.48	34.05	25.33	31.75	34.06	29.41	34.07	35.43	33.33	33.62	30.97	32.90
Cantor	28.18	44.44	31.47	21.40	29.37	31.88	26.05	30.37	33.07	28.57	25.00	26.55	29.00
ToT	33.41	47.47	37.93	24.45	33.33	36.23	30.25	34.81	37.01	33.33	31.03	33.63	33.80
VReST	34.32	51.52	36.21	26.20	33.33	36.96	35.29	36.30	36.22	33.33	31.03	33.63	34.60
VReST-Vote	34.32	52.53	35.78	31.00	34.92	39.86	36.97	34.07	40.16	35.71	30.17	32.74	35.70

Table 8: Accuracy scores (%) on the Validation set of CharXiv on the Qwen2.5-VL-3B-Instruct.

<image1>: 

<image2>: 

Question: A square piece of paper has been cut in three pieces. Two of them are in the picture on the right.

Ground Truth Answer: A

Sub-Q₁: What was the original shape of the paper?
Sub-A₁: The original shape of the paper was a square. **The answer is a square.**

Sub-Q₂: How many pieces is the square cut into?
Sub-A₂: The square is cut into three pieces. **The answer is three pieces.**

Sub-Q₃: What are the two visible pieces shown in the image?
Sub-A₃: The two visible pieces shown in the image are a small square and a more complex angular shape. **The answer is a small square and a complex angular shape.**

Sub-Q₄: Now we can answer the question: What is the third piece?
Sub-A₄: The third piece is the shape that completes the square, and based on the analysis of the pieces, the answer is the shape that fits with the given cut. The third piece **is likely option C** from the provided choices. The final answer is **option C**. ❌

Figure 6: Bad case of VReST frameworks.

sufficient granularity of problem decomposition. While VReST is capable of breaking down the problem into sub-questions, and each individual sub-question does not provide incorrect answers, the decomposition itself does not significantly contribute to the final answer. Moreover, the difficulty of the problem plays a critical role in this failure. The question belongs to the Level 5 category in the dataset, which is notably challenging and requires a more sophisticated and nuanced breakdown. In this case, the lack of depth in the decomposition did not equip the LVLm with the necessary insights to overcome the complexity of the task.

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.

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 $\frac{3}{4}$ full, calculate the total number of gallons of water they both have if Catherine's water heater is also full with water to $\frac{3}{4}$ 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.