AtomThink: A Slow Thinking Framework for Multimodal Mathematical Reasoning

Kun Xiang[1*], Zhili Liu[2*], Zihao Jiang[3*], Yunshuang Nie[1], Runhui Huang[4], Haoxiang Fan[5], Hanhui Li[1], Weiran Huang[3], Yihan Zeng[5], Jianhua Han[5], Lanqing Hong[5], Hang Xu[5], Xiaodan Liang[1†]
[1] Sun Yat-sen University [2] Hong Kong University of Science and Technology
[3] Shanghai Jiaotong University [4] University of Hong Kong [5] Huawei Noah's Ark Lab

Abstract

In this paper, we address the challenging task of multimodal mathematical reasoning by incorporating the ability of "slow thinking" into multimodal large language models (MLLMs). Contrary to existing methods that rely on direct or fast thinking, our key idea is to construct long chains of thought (CoT) consisting of atomic actions in a step-by-step manner, guiding MLLMs to perform complex reasoning. To this end, we design a novel AtomThink framework composed of three key modules: (i) a CoT annotation engine that automatically generates high-quality CoT annotations to address the lack of high-quality visual mathematical data; (ii) an atomic step fine-tuning strategy that jointly optimizes an MLLM and a policy reward model (PRM) for step-wise reasoning; and (iii) four different search strategies that can be applied with the PRM to complete reasoning. Additionally, we propose AtomMATH, a large-scale multimodal dataset of long CoTs, and an atomic capability evaluation metric for mathematical tasks. Extensive experimental results show that the proposed AtomThink significantly improves the performance of baseline MLLMs, achieving approximately 50% relative accuracy gains on MathVista and 120% on MathVerse. To support the advancement of multimodal slow-thinking models, we will make our code and dataset publicly available on https://github.com/Quinn777/AtomThink.

1. Introduction

Chain-of-thought (CoT) reasoning [34] has provided a novel scheme for large language models (LLMs) to tackle complex reasoning tasks. By utilizing a small number of specially designed instructions, CoT enables LLMs to generate intermediate reasoning steps, significantly enhancing performance on symbolic tasks such as mathematical problems and code writing [44].

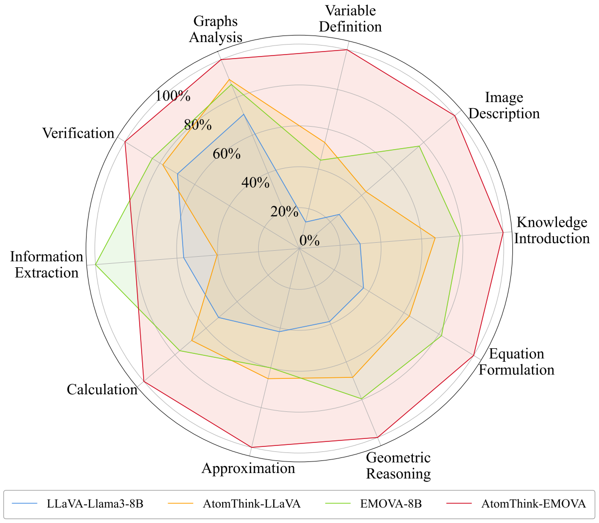


Figure 1. Atomic capability evaluation of different models. Existing open-source models exhibit significant shortcomings in capabilities such as variable definition, approximation and image description.

While CoT-based methods show clear improvements over direct predictions, they still rely heavily on greedy decoding strategies. More recently, the introduction of OpenAI's o1 [23] marks a substantial advancement in the ability of artificial intelligence systems to perform high-level reasoning. Unlike traditional models, o1 excels in solving complex problems by utilizing extended reasoning chains and adopting test-time scaling, i.e., "*slow thinking*". In addition to o1, several concurrent works have explored methods for incorporating slow thinking capabilities into open-source LLMs, such as Thought Trees [35] and Monte Carlo tree search (MCTS) based tree search techniques [6, 25, 30, 31]. The success of o1 and its variants demonstrate that incorporating slow thinking into LLMs significantly enhances their performance on com-

[*] These authors contributed equally to this work.
[†] Corresponding author. Email: xdliang328@gmail.com

plex, multi-step tasks, improving their overall problem-solving capabilities.

However, adopting the slow-thinking technique into multimodal large language models (MLLMs) is challenging, due to the increased data and computational resource demand for information modeling in visual tasks [24, 42]. Although many efforts have been conducted to alleviate this issue, such as incorporating interleaved image-text data [1], prompt engineering [20, 26], they are still confined to stimulating the inherent CoT capabilities of MLLMs, **without considering the quality of each intermediate step** in the reasoning chain. Hence, existing methods are hard to apply test-time scaling laws to guarantee their performance.

To validate the importance of the quality of each intermediate step in CoT, we first design a capability evaluation method to perform a fine-grained quality assessment of each atomic step generated by MLLMs. Here we define *an atomic step as the minimal prediction node in the slow thinking process*. Considering that humans may utilize distinct cognitive abilities for solving mathematical problems, we utilize one of the current most advanced LLMs, i.e., GPT-4o [21] to construct an ability set and estimate scores of atomic steps with outcome supervision. The results shown in Figure 1 indicate that the step quality of existing open-source models is significantly lower than that of GPT-4o, particularly in areas such as image recognition, variable definition, and calculation ability. This finding further motivates our focus on the capability gaps in existing models, prompting us to improve performance by enhancing the quality of atomic reasoning steps.

Therefore, to fully leverage the advantages of CoT and address the aforementioned challenges, we propose a full-process slow-thinking framework called AtomThink. AtomThink introduces a multimodal CoT annotation engine, an atomic step finetuning strategy, and policy searching strategies to generate high-quality atomic steps. It aims to enhance the decoding capabilities of MLLMs through careful training, combined with post-sampling search strategies to identify the optimal prediction nodes. To begin with, the proposed annotation engine is used to create a novel multimodal long CoT dataset including 26k high-level mathematical problems, 157k atomic-granularity steps, and 130k process supervision annotations. The construction of this dataset does not require manual labeling and effectively leverages existing short labels. Secondly, our atomic step finetuning strategy applies step-level masking to the training set, forcing our models to learn multi-turn self-dialogue ability and generate reasoning focused on individual inference actions. Thirdly, we explore different search strategies along both the path and step dimensions during the inference phase to find optimal prediction nodes. To validate the effectiveness of our method, we conduct extensive experiments on public datasets. We improved the accuracy of LLaVA-Llama3-8B on MathVista and MathVerse by 9.6% and 18.8%, respectively. With EMOVA (8B) as the base model, AtomThink achieved the highest accuracy of 40.5% on MathVerse, surpassing the cutting-edge GPT-4V.

In summary, our primary contributions are as follows:

- We introduce AtomThink, a comprehensive framework that guides MLLMs to focus on atomic step reasoning, which obtains consistent performance improvements across multiple baseline MLLMs.
- By designing an atomic capability evaluation based on outcome supervision, we reveal the capability distribution of MLLMs in generating each type of atomic step.
- A multimodal long CoT dataset specifically focused on multimodal mathematical tasks, AtomMATH, is first introduced.

2. Related Work

Slow Thinking in Multimodal Reasoning Tasks Complex reasoning tasks such as mathematical computation and code programming have long been challenging for MLLMs [15, 36, 44]. Some prior work has approached this issue from the perspective of prompt engineering, encouraging models to generate Chain-of-Thought(CoT), which is widely believed to enhance model's reasoning [33, 34]. They carefully modify the input distribution to enable the model to mimic human step-by-step output without finetuning parameters. Other recent studies have explored understanding visual ambiguity by introducing multi-turn chain-of-thoughts [20]. Shao et al. [26] have considered incorporating additional visual tokens into CoTs, such as object regions and precise localization. However, due to the lack of multimodal process supervision data, current works have not explored reward model-based search strategies, which are widely used in LLMs [3, 12, 28, 29, 38].

Long CoT Annotation for Mathematical Data The introduction of slow thinking relies heavily on the availability of high-quality step-level annotations. In 2023, Lightman et al. [11] constructed a process supervision dataset composed of extensive human annotations, which has been widely used for mathematical reasoning. Recent advancements have focused on automating the data acquisition process, allowing models to generate their own CoT. Techniques like Quiet-STaR [39] have demonstrated how self-generated reasoning can enhance model performance without requiring manually labels. Moreover, some methods based on Monte Carlo estimation have automated the process data collection, but they also introduce additional computational cost [19, 32]. In multimodal domain, MAVIS [41], a dataset consisting of 834k visual math problems annotated with short CoT, has been proposed. Other studies have distilled reasoning processes from short answers

[42]. However, these machine-generated annotations are often too brief and challenging to segment semantically.

3. Method

In this section, we present the details of AtomThink for promoting MLLMs for mathematical reasoning with slow thinking. As shown in Figure 2, AtomThink consists of three key components, including a multimodal CoT annotation engine (Sec. 3.1), atomic step fine-tuning (Sec. 3.2), and policy searching (Sec. 3.3). The annotation engine is designed to efficiently generate long CoTs to address data scarcity. With sufficient data, we fine-tune MLLMs and train a process reward model (PRM) for incorporating slow thinking ability into models. Furthermore, we explore four different path-wise and step-wise strategies for policy searching, allowing the fine-tuned MLLM to ensure that each decision made during its inference contributes to the overall accuracy and consistency of reasoning. Finally, we propose an atomic capability evaluation metric in Sec. 3.4 to measure the reasoning quality of models.

Source	Meta Samples	AMATH-SFT	AMATH-PRM
CLEVR	1929	11.2k	25k
Geometry3K	1201	11.1k	15.6k
MAVIS	3654	17.7k	30.5k
TabMWP	2463	15.7k	25.7k
GeomVerse	1347	9.9k	17k
MathV360K	10157	53.5k	24.8k
MMMU	76	0.6k	1.2k
GeoQA+	2082	19.5k	0
IconQA	3199	18.1k	0
Total	26108	157k	130k

Table 1. Data composition of AtomMATH.

Data	GPT Score	Avg. Length
PRM800k	84.1	1245.4
Direct	1.5	3.6
CoT	79.6	670.5
AtomMATH(Ours)	89.4	849.8

Table 2. Comparison of different data styles. Our AtomMATH achieves the highest GPT-4o preference score.

3.1. Multimodal CoT Annotation Engine

Guiding MLLMs toward deep reasoning requires a substantial amount of high-quality CoT data. However, in the field of visual mathematics, the scarcity of publicly available datasets presents a considerable challenge. To overcome this, we develop an automated data engine capable of generating step-by-step long CoTs, resulting in our own atomic mathematical problem dataset, dubbed AtomMATH. Specifically, our data engine introduces a dynamic prompting strategy and a semantic-level augmentation strategy to produce multi-step reasoning paths.

Dynamic Prompting Strategy. To overcome the computational cost limitations associated with previous methods that relied on manual annotation or process supervision, we explore the possibility of driving existing models to autonomously generate high-quality reasoning data through simple prompting. Inspired by recent research [9] on using prompting strategies to improve the reasoning capabilities of LLMs, we propose a dynamic prompt strategy for generating atomic inference steps. Specifically, our strategy drives LLMs to iteratively construct state-reasoning paths. Each path node represents a reasoning step and encompasses the previous stage, the current state, and a possible action. The possible action includes continuing reasoning, verifying and drawing conclusion, which is determined by LLM itself. Unlike previous methods such as OmegaPRM [19] and Math-Shepherd [32] that generate a whole reasoning tree at once, our approach implicitly integrates the search over the step dimension into existing reasoning process through prompt engineering. For each problem instance, only a single valid path is explored, eliminating the need for additional process supervision computation.

Short CoT Augmentation. To fully leverage existing short CoT annotations of VQA datasets, we also employ LLMs to atomize and augment these annotations. An example of short CoT augmentation is provided in the supplemental material. This approach allows us to semantically segment an original reasoning process into multiple discrete steps, and focus on solving a single atomic problem at each stage of the reasoning process, thereby ensuring the clarity and precision of our model.

AtomMATH Dataset. We sample mathematical data from Geo3k [17], Mathv360k [27], MMMU-dev [37], TabMWP [16], CLEVR [7], Geomverse-Cauldron [8] and MAVIS [41]. For Geomverse and MAVIS, we conduct short CoT augmentation, while the rest are generated by dynamic prompts to produce multi-step reasoning. Both short CoT augmentation and dynamic prompting are implemented by GPT-4o in this paper. After generating long CoTs, we also use GPT-4o to double-check the answers and remove rollouts with incorrect responses. To enable our model to learn atomic step-based reasoning patterns, we progressively mask each node along its reasoning path to generate 157k atomic steps. We refer to this database as AMATH-SFT. Meanwhile, we sampled approximately 65k examples with correct steps from AMATH-SFT, and generated negative samples using GPT-4o to serve as PRM training data. Table 1 illustrates the distribution of our data. In Table 2, we also evaluate the quality in a subset of 500 AtomMATH

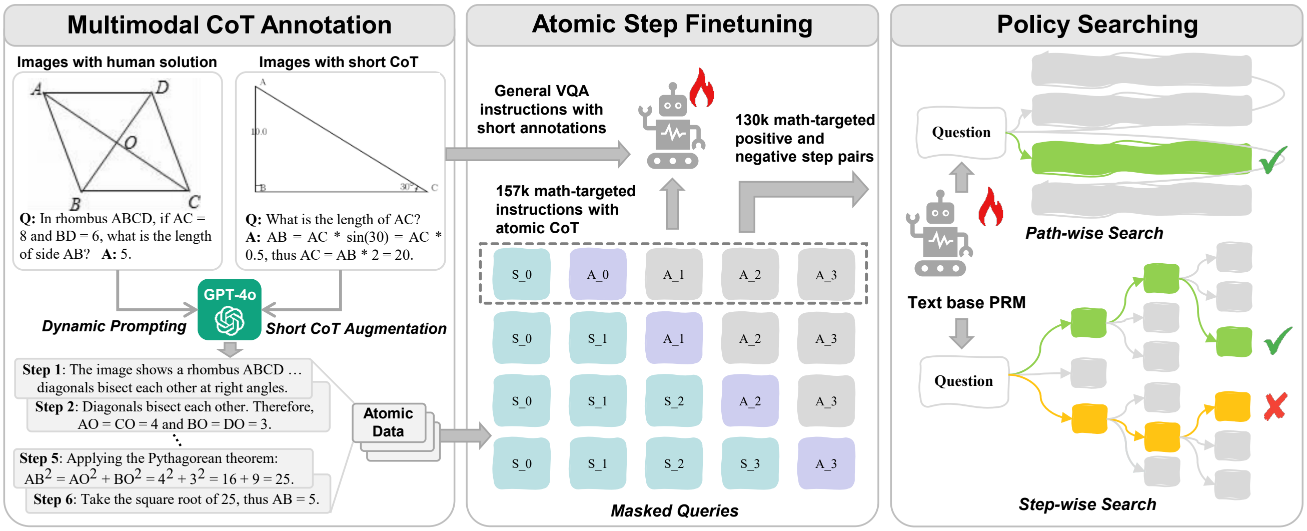

Multimodal CoT Annotation
Images with human solution
Images with short CoT
A
D
O
B
C
Q: In rhombus ABCD, if AC = 8 and BD = 6, what is the length of side AB? A: 5.
A
10.0
B
30°
C
Q: What is the length of AC?
A: AB = AC * sin(30) = AC * 0.5, thus AC = AB * 2 = 20.
GPT-4o
Dynamic Prompting
Short CoT Augmentation
Step 1: The image shows a rhombus ABCD … diagonals bisect each other at right angles.
Step 2: Diagonals bisect each other. Therefore, AO = CO = 4 and BO = DO = 3.
Step 5: Applying the Pythagorean theorem: AB2 = AO2 + BO2 = 42 + 32 = 16 + 9 = 25.
Step 6: Take the square root of 25, thus AB = 5.
Atomic Data
Atomic Step Finetuning
General VQA instructions with short annotations
130k math-targeted positive and negative step pairs
157k math-targeted instructions with atomic CoT
S_0
A_0
A_1
A_2
A_3
S_0
S_1
A_1
A_2
A_3
S_0
S_1
S_2
A_2
A_3
S_0
S_1
S_2
S_3
A_3
Masked Queries
Policy Searching
Question
Path-wise Search
Text base PRM
Question
Step-wise Search

Figure 2. The overview of AtomThink framework. We automatically annotate the open-source data with CoT to generate atomic steps for fine-tuning and PRM training. During inference, step-wise or path-wise searching strategies can be applied to find optimal policies.

samples with GPT-4o. Result shows that our method exhibits longer information content than general CoT. Even compared to the PRM800k with golden annotations, our data obtained a higher preference score.

3.2. Atomic Step Fine-Tuning

To fully exploit MLLMs for addressing multi-modal mathematical problems, we conduct fine-tuning with atomic stepwise reasoning. Particularly, this process includes fine-tuning the MLLM on our AtomMATH dataset and learning the PRM to estimate reward scores during the inference.

MLLM Fine-Tuning. To transfer MLLM to step-wise mathematical reasoning, we first fine-tune it within the framework of Markov decision process (MDP) learning. Specifically, we consider the reasoning process of MLLM as an MDP, which can be formulated as $M = (V, S, A, R, \pi)$. Here, V denotes the vocabulary, S represents historical reasoning steps, and A corresponds to next atomic step predicted by MLLM. $\pi(a|s)$ represents the probability of selecting an action $a \in A$ conditioned on a state $s \in S$, which is estimated by PRM to guide reasoning process. Hereby we can adopt the visual instruction tuning technique [14] to fine-tune MLLM.

PRM Training. In a slow thinking process, reasoning is carried out step by step, where each atomic step provides an intermediate conclusion. We train the PRM to implement $\pi(a|s)$ and provide feedback for every step, allowing MLLMs to refine and improve its reasoning. Formally, given the description of mathematical problem q, for an arbitrary step $t \geq 1$, the PRM predicts a probability p_t of selecting an action a given the previous states $s_{1:t-1}$ as follows:

$$p_t(a) = \text{PRM}\left([q, s_{1:t-1}], a\right). \tag{1}$$

We propose to train the PRM by minimizing the following binary cross-entropy loss:

$$\mathcal{L}_{PRM} = \sum_{t=1}^{T} y_t(a) \log p_t(a) + (1 - y_t(a)) \log(1 - p_t(a)), \tag{2}$$

where $y_t(a)$ denotes the ground-truth CoT annotation that $y_t = 1$ if the action a is selected, otherwise $y_t(a) = 0$. T is the maximum number of steps. Note that we omit to enumerate all possible actions in Eq. (2) for the concise presentation. After selecting the action at the current step a_t, we concatenate it with the previous states to construct s_t, i.e., $s_t = s_{1:t-1} \cup a_t$.

In this subsection, we perform atomic step fine-tuning on the AtomMATH (including A-PRM and A-SFT subsets) and PRM800k dataset [11]. Moreover, we incorporate image captions into the generation of long CoT data, thus we can alleviate the expensive computation burden of image understanding in MLLMs and focus on texts for supervised fine-tuning. Therefore, we post-train an LLM based on Math-psa [31] to evaluate the consistency of atomic texts and supervise fine-tuning.

3.3. Action Search with PRM

With the fine-tuned MLLM capable of atomic step reasoning and the well-trained PRM providing feedback, we can now begin the reasoning process. As there are many search strategies to generate candidate actions, we categorize the existing strategies into path-wise searching and step-wise searching and explore them in our AtomThink framework.

Path-wise Search. In path-wise searching, we build upon prior work [28, 31] by parallel sampling multiple paths and aggregating scores to find optimal solutions. We investigate the following two strategies:

- **Majority Voting:** It combines multiple reasoning paths by selecting the most frequent outcome across them. It assumes that the consensus across different paths is more likely to lead to the correct answer.
- **Best-of-N:** Given a generative MLLM, the best-of-N sampling method generates C candidate rollouts simultaneously and selects the solution with the highest score. The evaluation of candidate reasoning processes is determined by the PRM, which employs three aggregation methods to map the dense scores to the overall value of the entire path: **1) The worst action:** Compare the worst action among all candidate rollouts. It penalizes solutions with any weak action and is used to search a reasoning that is sensitive to errors. **2) The last action:** The score is derived from the prediction of the final answer in inference. **3) Average score:** It is calculated by averaging the rewards of all the actions in a chain. The explainability and consistency of intermediate reasoning are emphasized here as important as the outcome.

Step-wise Search. Searching strategies of this type start with an initial path and incrementally expand the sampling space for each atomic action. Beam search and greedy strategies are applied to prune branches with low quality.

- **Greedy Algorithm:** It focuses on making the locally optimal choice at each step of the reasoning process. It selects the best immediate action (step) based on the current state, without considering future consequences.
- **Beam Search:** It explores multiple branches at each action and maintains a fixed number of top candidates for each stage of reasoning. It balances between exploring different paths and exploiting the most promising ones.

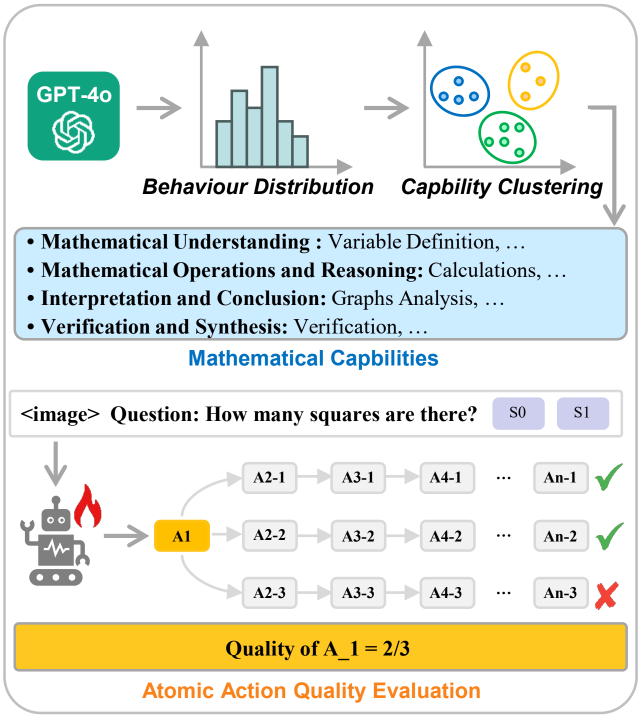

GPT-4o
Behaviour Distribution
Capbility Clustering
• Mathematical Understanding : Variable Definition, …
• Mathematical Operations and Reasoning: Calculations, …
• Interpretation and Conclusion: Graphs Analysis, …
• Verification and Synthesis: Verification, …
Mathematical Capbilities
<image> Question: How many squares are there?
S0
S1
A1
A2-1
A3-1
A4-1
···
An-1
A2-2
A3-2
A4-2
···
An-2
A2-3
A3-3
A4-3
···
An-3
Quality of A_1 = 2/3
Atomic Action Quality Evaluation

Figure 3. Atomic capability evaluation. The capability categories are derived from the clustering of GPT-4o's behavior. By sampling each atomic step and evaluating the accessibility of the results, we assign a soft label that represents quality of an atomic step.

3.4. Atomic Capability Evaluation

Similar to human problem-solving processes, a CoT may involve the use of multiple reasoning abilities. However, traditional CoT methods do not focus on the quality of individual reasoning steps or provide fine-grained analyses of the underlying abilities. To address this gap, we have developed an atomic capability evaluation strategy, offering a new analytical perspective for slow thinking.

Our evaluation method aims to assess the mathematical capabilities of a target model from various perspectives, such as understanding, operations, and certifications. To this end, we first need to construct a canonical set of capabilities. As shown in Figure 3, we collect the behavior distribution of GPT-4o on MathVerse [43] and perform clustering, yielding clusters that each of them represents certain abilities utilized by high-level intelligent models in solving mathematical problems. Afterward, we evaluate an atomic action a predicted by the target model based on outcome supervision, namely, we consider that the quality of an action can be reflected by the probability of it leading to a correct answer. This can be formulated as the following soft estimation with K rounds of outcome supervision [11]:

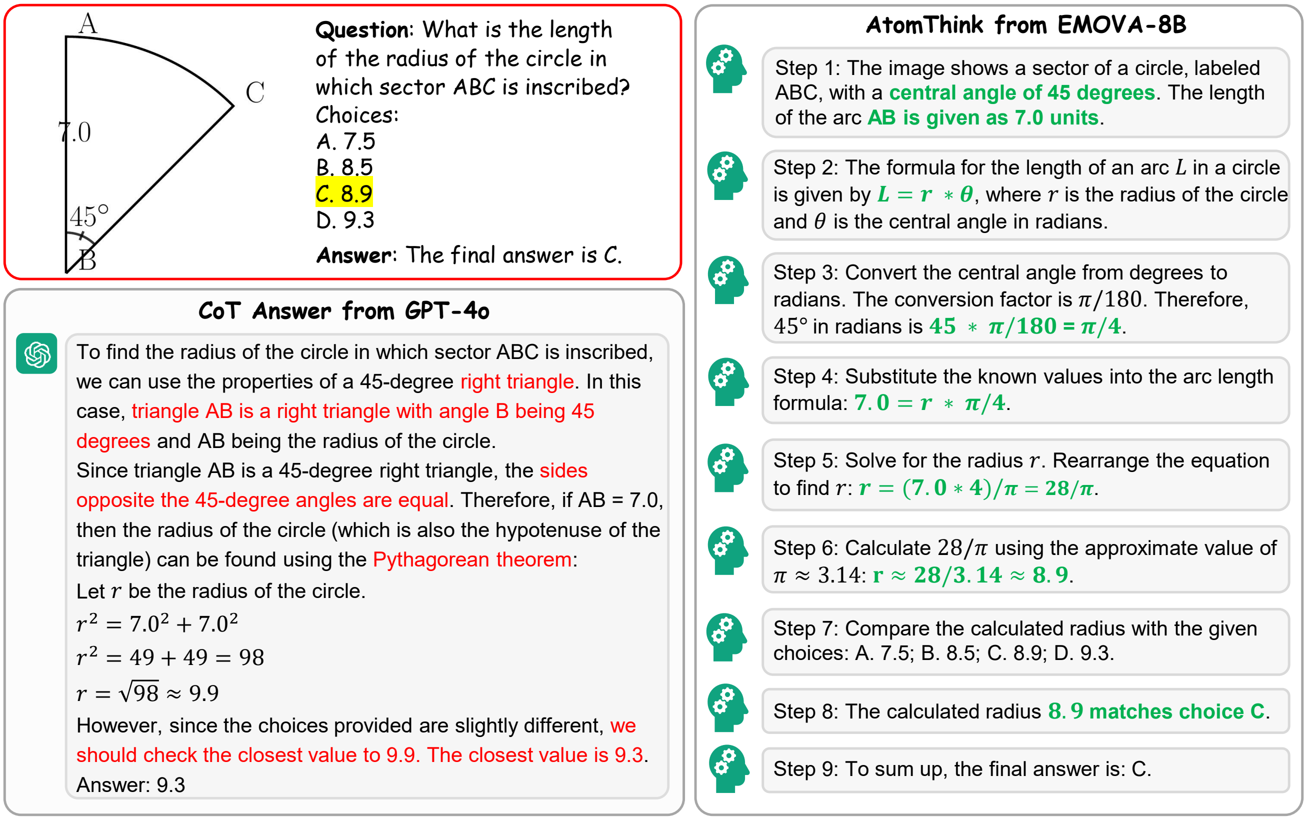
$$c_{soft}(a) = \frac{\sum_{k=1}^{K} [\![\hat{a}_k \text{ is correct}]\!]}{K}, \tag{3}$$

where $\hat{a}_k$ denotes the final answer predicted in the k-th round and $[\![\cdot]\!]$ is the Iverson bracket. Consequently, we can evaluate all the actions of the target model and map them back to the capability set, to evaluate the mathematical capabilities of the target model thoroughly.

4. Experiment

4.1. Setup

Baselines. Our experiments utilize two open-source MLLMs, including LLaVA-Llama3-8B [14] and EMOVA-8B [4]. We fine-tune only the parameters of their language



Question: What is the length of the radius of the circle in which sector ABC is inscribed?
Choices:
A. 7.5
B. 8.5
C. 8.9
D. 9.3

Answer: The final answer is C.

CoT Answer from GPT-4o

To find the radius of the circle in which sector ABC is inscribed, we can use the properties of a 45-degree right triangle. In this case, triangle AB is a right triangle with angle B being 45 degrees and AB being the radius of the circle.
Since triangle AB is a 45-degree right triangle, the sides opposite the 45-degree angles are equal. Therefore, if AB = 7.0, then the radius of the circle (which is also the hypotenuse of the triangle) can be found using the Pythagorean theorem:
Let r be the radius of the circle.
$r^2 = 7.0^2 + 7.0^2$
$r^2 = 49 + 49 = 98$
$r = \sqrt{98} \approx 9.9$
However, since the choices provided are slightly different, we should check the closest value to 9.9. The closest value is 9.3.
Answer: 9.3

AtomThink from EMOVA-8B

Step 1: The image shows a sector of a circle, labeled ABC, with a central angle of 45 degrees. The length of the arc AB is given as 7.0 units.

Step 2: The formula for the length of an arc L in a circle is given by $L = r * \theta$, where r is the radius of the circle and θ is the central angle in radians.

Step 3: Convert the central angle from degrees to radians. The conversion factor is $\pi/180$. Therefore, 45° in radians is $45 * \pi/180 = \pi/4$.

Step 4: Substitute the known values into the arc length formula: $7.0 = r * \pi/4$.

Step 5: Solve for the radius r. Rearrange the equation to find r: $r = (7.0 * 4)/\pi = 28/\pi$.

Step 6: Calculate $28/\pi$ using the approximate value of $\pi \approx 3.14$: $r \approx 28/3.14 \approx 8.9$.

Step 7: Compare the calculated radius with the given choices: A. 7.5; B. 8.5; C. 8.9; D. 9.3.

Step 8: The calculated radius 8.9 matches choice C.

Step 9: To sum up, the final answer is: C.

Figure 4. A case study of AtomThink CoT and GPT-4o generated CoT. Red and green characters denote incorrect and correct responses, respectively. Our model exhibits fewer hallucinations and stronger reasoning abilities as it focuses on the quality of atomic steps. Moreover, our model automatically decomposes the reasoning process semantically, leading to improved readability.

models and projectors with learning rates of 2e-5 and 2e-6, respectively, and a batch size of 128. We select nine cutting-edge MLLMs for comparison, including OpenAI's o1 [23], 4o [21], and 4v [22], as well as LLava-NeXT-34B [13], InternLM-XComposer2 [41], Qwen-VL-Plus [2], LLaVA-7B [14], G-LLaVA-7B [5], and MAVIS-7B [41].

Datasets. For **LLaVA-Llama3-8B**, we use LLaVA-665k [14] for supervised fine-tuning (SFT) as a baseline. Additionally, in **LLaVA w/. Formatted** and **EMOVA w/. Formatted**, we transfer the source data of AtomMATH into an aligned CoT format for incremental training, ensuring a fair comparison without introducing bells and whistles. For EMOVA-8B, we downsampled its publicly available SFT data [4] to obtain a basic post-training dataset containing about 200k samples. For models with AtomThink, the AMATH-SFT dataset introduced in Section 3.1, is incorporated to introduce atomic reasoning capabilities.

Evaluation Setting. We evaluated the performance of our method on MathVista [18], a publicly available benchmark encompassing both general-targeted and mathematics-targeted domains. Additionally, to assess the model's ability to interpret mathematical graphs, we use a more challenging multimodal benchmark, MathVerse [43] for further evaluation. It contains five categories including Text Lite (TL), Text Dominant (TD), Vision Intensive (VI), Vision Dominant (VD), Vision Only (VO).

Out evaluations include four inference settings, including **Direct**, **CoT**, **Quick Think**, and **Slow Think**. In the **Direct** setting, we prompt the model to generate a concise final answer. In **CoT**, the model is instructed to answer the question through step-by-step reasoning. For the *Direct* and **CoT** evaluations, we use prompts from lmms-eval [10, 40]. Our AtomThink-models support two additional settings: **Quick Think** and **Slow Think**. In **Quick Think**, our models follow a single, atomic reasoning path based purely on their learned policies, without employing any supplementary search strategies. In **Slow Think**, enhanced by the PRM, we utilize beam search with beam width of 2 and temperature of 1.0, encouraging our models to engage in more extensive reasoning.

		MathVista			MathVerse					
Model	**Inference**	**General**	**Math**	**Total**	**TL**	**TD**	**VI**	**VD**	**VO**	**Total**
Random Choice	-	-	-	17.9	12.4	12.4	12.4	12.4	12.4	12.4
Human	-	-	-	-	70.9	71.2	61.4	68.3	66.7	66.7
OpenAI o1	Slow Think*	-	-	73.9	-	-	-	-	-	-
GPT-4o	CoT	-	-	63.8	-	-	-	-	-	-
GPT-4V	CoT	-	-	49.9	56.6	63.1	51.4	50.8	50.3	54.4
LLaVA-NeXT-34B	Direct	-	-	46.5	25.5	33.8	23.5	20.3	15.7	23.8
InternLM-XComposer2	Direct	-	-	57.6	17.0	22.3	15.7	16.4	11.0	16.5
Qwen-VL-Plus	Direct	-	-	43.3	11.1	15.7	9.0	13.0	10.0	11.8
LLaVA-1.5-13B	Direct	-	-	27.6	15.2	19.4	16.8	15.2	11.3	15.6
G-LLaVA-7B	Direct	-	-	53.4	20.7	20.9	17.2	14.6	9.4	16.6
MAVIS-7B	Direct	-	-	-	29.1	41.4	27.4	24.9	14.6	27.5
LLaVA-Llama3-8B	Direct	34.1	25.6	29.5	16.0	19.3	16.4	13.1	15.0	15.9
LLaVA w/. Formatted	CoT	30.2	22.9	26.3	14.3	18.4	15.7	10.0	7.7	13.2
AtomThink-LLaVA	Direct	34.4	27.2	30.5	16.0	19.3	16.2	13.1	15.0	15.9
AtomThink-LLaVA	Quick Think	**36.9**	**37.0**	**36.6**	**22.2**	**26.6**	**24.1**	**20.9**	**17.9**	**22.4**
AtomThink-LLaVA	Slow Think	**36.5**	**41.3**	**39.1**	**36.1**	**42.4**	**30.0**	**36.8**	**28.6**	**34.7**
EMOVA-8B-200k	Direct	52.4	51.1	51.7	34.4	39.0	33.4	30.1	23.5	32.1
EMOVA w/. Formatted	CoT	30.9	31.3	31.1	26.5	36.5	25.3	20.4	19.8	25.7
AtomThink-EMOVA	Direct	53.9	52.4	53.1	33.6	39.0	33.8	28.0	24.4	31.8
AtomThink-EMOVA	Quick Think	48.7	**54.4**	**51.8**	**36.5**	**42.4**	**34.1**	**32.9**	**29.7**	**35.1**
AtomThink-EMOVA	Slow Think	48.9	**57.0**	**53.3**	**42.1**	**51.5**	**39.0**	**36.7**	**33.1**	**40.5**

Table 3. Comparison of accuracy with state-of-the-art methods on MathVista and MathVerse. Our AtomThink LLaVA outperforms the baseline in all sub-tasks across two benchmarks, achieving an average improvement of 14.2%. Meanwhile, AtomThink EMOVA, with only 8B parameters, surpasses the larger LLaVA-NEXT-34B and even is comparable to GPT-4V.

4.2. Main Results

Comparison with existing MLLMs. In Table 3, our AtomThink framework is applied to train LLaVA-Llama3-8B and EMOVA-8B, yielding consistent performance improvements over the original models. When combined with PRM, AtomThink-EMOVA achieves a new state-of-the-art on MathVerse, surpassing GPT-4o and narrowing the gap between MLLMs and human performance. On MathVista, it also achieves performance close to that of GPT-4o. These results demonstrate the framework's strong generalization capability and practical usability.

Quick Think with Intuition. Unlike traditional CoT methods, Quick Think generates a stepwise reasoning path through multi-turn conversations, bypassing the need for an additional verifier. This approach offers a computational advantage over Slow Think and highlights the model's intuitive reasoning capabilities. For LLaVA-Llama3-8B, our AtomThink framework surpasses the baseline model, achieving approximately a 10% improvement on MathVista [18] and a 19% improvement on MathVerse [43]. For AtomThink-EMOVA, Quick Think achieved a score of 38.3% on MathVerse, outperforming existing open-source MLLMs. These results demonstrate that when a model possesses atomic reasoning capabilities, it can leverage rapid intuition to perform more accurate mathematical reasoning.

LLM Effectively Supervise Visual Reasoning Processes. Previous work has shown that process supervision reward models are effective in evaluating intermediate reasoning steps, though these methods have been primarily applied within the domain of language models. We fine-tuned an LLM with A-MATH-PRM and applied it for test-time scaling. As shown in the table, AtomThink-EMOVA, when utilizing PRM with beam search, achieved an additional 2% improvement on MathVista [18] compared to Quick Think. In MathVerse [43], it even outperformed the closed-source model GPT-4V by 1%. Additionally, increasing test-time scaling in LLAVA resulted in substantial improvements, positioning it well above its sibling model, LLAVA-1.5-13B.

We find that even when the reasoning process heavily relies on visual dominant inputs, our models can avoid taking incorrect paths by improving text decoding. On the one hand, it is attributed to the AtomThink training process, which encourages MLLM to first understand image before reasoning. On the other hand, it also confirms the effectiveness of test-time extension in multimodal tasks.

Trade-off between General and Math Ability. Similar to the conclusions reported in o1, we observe that MLLMs become weaker on general tasks that rely on world knowledge during deep contemplation, demonstrating a trade-off between higher-level reasoning and direct thinking. For instance, LLaVA-Llama3-8B presents a decline in accuracy of 7% compared to the baseline on the general subset of MathVista, while EMOVA experiences a 17% reduction. However, after applying PRM-based action search, both models are able to narrow this generalization gap and improve accuracies by 4% and 16%, respectively.

4.3. Atomic Ability Analysis

We first cluster the reasoning behaviors of GPT-4o into a set of capabilities $\mathcal{S}$, including Approximation, Verification, Calculation, Variable Definition, Geometric Reasoning, Conclusion Drawing, Graphs Analysis, Equation Formulation, Image Description, Knowledge Introduction, Information Extraction, and Formula Derivation. Using queries in MathVerse [43], we constructed 500 current states as s_i with high-quality responses generated by GPT-4o. Subsequently, soft estimations of atomic actions are mapped to $\mathcal{S}$ for analysis.

Ability Analysis. Figure 3 illustrates the distribution of atomic behaviors and capability differences between LLAVA-llama3-8b and EMOVA-8B with their AtomThink-versions. The analysis reveals that AtomThink-Model generally outperforms baseline across most abilities, demonstrating higher scores in areas such as Image Description and Verification. It suggests that our model is capable of more accurate analysis of visual information and demonstrates a degree of self-checking and reflective capability.

4.4. Comparison with g1

In Figure 5, we compare AtomThink with the state-of-the-art open-source inference strategy, g1[1], which employs dynamic prompting to make model focus on single step reflection. In GPT-4o, direct application of g1 for multi-turn reasoning yields a greater improvement over Chain-of-Thought, particularly in numeric and geometric tasks. However, due to the reliance on the inherent reasoning capabilities of large-scale language models, its performance significantly degrades on smaller models such as EMOVA-8B and LLaVA-Llama3-8B. In contrast, our AtomThink framework consistently enhances the performance of these MLLMs.

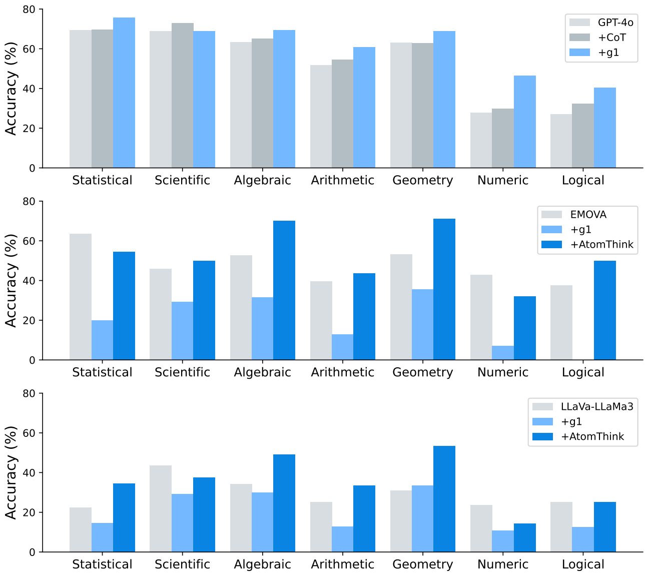


Figure 5. Comparison to CoT and g1 in MathVista subsets. In contrast to the declining trend observed in g1, AtomThink outperforms the baseline across most subsets.

4.5. Action Search Ablation

In Table 4, we evaluate the impact of direct output, path-wise search and action-wise search strategies on MathVista [18] and MathVerse [43] using a subset of 200 samples from each. Results show that even without additional computation, AtomThink-EMOVA's direct prediction accuracy outperforms the original, with improvements of 1.3%, 1.52%, and 2.4%, respectively. The path-wise search method, BoN-Avg, achieves the highest accuracy of 58.68% on the MathVista [18] mathematical tasks, although it experienced a drop on general problems. Meanwhile, both greedy algorithm and beam search show balanced performance across all benchmarks, with the generalization gap between math and general tasks being notably smaller than that of path-wise search.

Model	Method	MathVista-M	MathVista-G	MathVerse
EMOVA-200k	Direct	51.1	52.4	33.3
AtomThink	Direct Quick Think	52.4 54.2	53.9 46.7	35.7 38.1
w/. Path-wise	Majority Voting BoN-Last BoN-Avg BoN-Min	48.8 51.2 58.7 53.7	49.4 46.8 40.5 53.2	39.0 41.3 38.7 40.0
w/. Step-wise	Greedy Beam Search	46.3 57.1	45.6 53.2	38.3 45.3

Table 4. Ablation study on Path-wise and step-wise search. The results show that both Best-of-N-Min(BoN-Min) and Beam Search exhibit consistent performance improvements.

Limitation. Due to the limitations in computing infras-

[1] https://github.com/bklieger-groq/g1

tructure, we are unable to validate our method on larger MLLMs. Additionally, despite undergoing small-scale manual review, our dataset still lacks step-level golden answers, which may introduce noise into training.

5. Conclusion

This paper introduces atom thinking capabilities to MLLMs for solving visual mathematics problems. We release a high-quality, human-free annotated long-CoT dataset, AtomMATH, consisting of 157k atomic reasoning steps and 130k corresponding annotations. Furthermore, we propose AtomThink, a novel framework that focuses on the quality of atomic steps. The experimental results demonstrate that our method consistently enhances the model's diverse behaviors during the problem-solving process, leading to improved reasoning performance across various multimodal mathematical tasks. This work paves the way for developing generalized slow-thinking models.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023. 6

[3] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. 2

[4] Kai Chen, Yunhao Gou, Runhui Huang, Zhili Liu, Daxin Tan, Jing Xu, Chunwei Wang, Yi Zhu, Yihan Zeng, Kuo Yang, et al. Emova: Empowering language models to see, hear and speak with vivid emotions. *arXiv preprint arXiv:2409.18042*, 2024. 5, 6

[5] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. G-llava: Solving geometric problem with multi-modal large language model, 2023. 6

[6] Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. Interpretable contrastive monte carlo tree search reasoning. *arXiv preprint arXiv:2410.01707*, 2024. 1

[7] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 3

[8] Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. Geomverse: A systematic evaluation of large models for geometric reasoning. In *CVPRW*. 3

[9] Benjamin Klieger. g1: Using llama-3.1 70b on groq to create o1-like reasoning chains, 2024. 3

[10] Bo Li, Peiyuan Zhang, Kaichen Zhang, Xinrun Du Fanyi Pu, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Accelerating the development of large multimodal models, 2024. 6

[11] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*. 2, 4, 5

[12] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. 2

[13] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 6

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2024. 4, 5, 6

[15] Wentao Liu, Hanglei Hu, Jie Zhou, Yuyang Ding, Junsong Li, Jiayi Zeng, Mengliang He, Qin Chen, Bo Jiang, Aimin Zhou, et al. Mathematical language models: A survey. *arXiv preprint arXiv:2312.07622*, 2023. 2

[16] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *ICLR*. 3

[17] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *ACL*, pages 6774–6786, 2021. 3

[18] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *arXiv e-prints*, pages arXiv–2310, 2023. 6, 7, 8

[19] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint:2406.06592*, 2024. 2, 3

[20] Minheng Ni, Yutao Fan, Lei Zhang, and Wangmeng Zuo. Visual-o1: Understanding ambiguous instructions via multi-modal multi-turn chain-of-thoughts reasoning. *arXiv preprint arXiv:2410.03321*, 2024. 2

[21] OpenAI. Gpt-4o system card, . 2, 6

[22] OpenAI. Gpt-4v(ision) system card, . 6

[23] OpenAI. Openai o1 system card, . 1, 6

[24] Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, et al. Cogcom: Train large vision-language models diving into details through chain of manipulations. *arXiv preprint arXiv:2402.04236*, 2024. 2

[25] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. O1 replication journey: A strategic progress report–part 1. *arXiv preprint arXiv:2410.18982*, 2024. 1

[26] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Unleashing chain-of-thought reasoning in multi-modal language models. *arXiv preprint arXiv:2403.16999*, 2024. 2

[27] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See Kiong Ng, Lidong Bing, and Roy Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. pages 4663–4680, 2024. 3

[28] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. 2, 5

[29] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. 2

[30] Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. Q*: Improving multi-step reasoning for llms with deliberative planning. *arXiv preprint arXiv:2406.14283*, 2024. 1

[31] Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, et al. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*, 2024. 1, 4, 5

[32] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *ACL*, pages 9426–9439, 2024. 2, 3

[33] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 2

[34] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 1, 2

[35] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[36] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 2

[37] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, pages 9556–9567, 2024. 3

[38] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022. 2

[39] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024. 2

[40] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. 6

[41] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Yichi Zhang, Ziyu Guo, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, et al. Mavis: Mathematical visual instruction tuning. *arXiv e-prints*, pages arXiv–2407, 2024. 2, 3, 6

[42] Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. Improve vision language model chain-of-thought reasoning. *arXiv preprint arXiv:2410.16198*, 2024. 2, 3

[43] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2025. 5, 6, 7, 8

[44] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 1, 2