

# Can Atomic Step Decomposition Enhance the Self-structured Reasoning of Multimodal Large Models?

Kun Xiang<sup>1\*</sup>, Zhili Liu<sup>2\*</sup>, Zihao Jiang<sup>3\*</sup>, Yunshuang Nie<sup>1</sup>, Kaixin Cai<sup>1</sup>, Yiyang Yin<sup>1</sup>, Runhui Huang<sup>4</sup>, Haoxiang Fan<sup>1</sup>, Hanhui Li<sup>1</sup>, Weiran Huang<sup>3</sup>, Yihan Zeng<sup>5</sup>, Yu-Jie Yuan<sup>5</sup>, Jianhua Han<sup>5</sup>, Lanqing Hong<sup>5</sup>, Hang Xu<sup>5</sup>, Xiaodan Liang<sup>1†</sup>

<sup>1</sup> Sun Yat-sen University <sup>2</sup> Hong Kong University of Science and Technology

<sup>3</sup> Shanghai Jiaotong University <sup>4</sup> University of Hong Kong <sup>5</sup> 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). Our core idea is that different levels of reasoning abilities can be combined dynamically to tackle questions with different complexity. To this end, we propose a paradigm of Self-structured Chain of Thought (SCoT), which is composed of minimal semantic atomic steps. Different from existing methods that rely on structured templates or free-form paradigms, our method can not only generate cognitive CoT structures for various complex tasks but also mitigates the phenomenon of overthinking. To introduce structured reasoning capabilities into visual understanding models, we further design a novel AtomThink framework with four key modules, including (i) a data engine to generate high-quality multimodal reasoning paths; (ii) a supervised fine-tuning process with serialized inference data; (iii) a policy-guided multi-turn inference method; and (iv) an atomic capability metric to evaluate the single step utilization rate. We conduct extensive experiments to show that the proposed AtomThink significantly improves the performance of baseline MLLMs, achieving more than 10% average accuracy gains on MathVista and MathVerse. Compared to state-of-the-art structured CoT approaches, our method not only achieves higher accuracy but also improves data utilization by 5 times and boosts inference efficiency by 85.3%. Our code is now public available in <https://github.com/Quinn777/AtomThink>.*

## 1. Introduction

Chain-of-Thought (CoT) reasoning [42] 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 unstructured reasoning steps to enhance their performance on complex tasks. Moreover, the introduction of OpenAI’s o1 [30] then 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”.

More recently, numerous studies have ventured to hypothesize about their reasoning paradigms [11, 32, 37, 45]. Certain endeavors, such as LLaVA-CoT [43] and LlamaV-o1 [36], have posited structured methodologies employing fixed modules to drive reasoning. However, these methods require manually designed thinking templates, which limits the diversity of their reasoning behaviors in multimodal complex problems. In contrast to structured approaches, the advent of DeepSeek-R1 [12] has redirected attention towards unstructured reasoning. While unstructured chains of thought more closely mirror human cognitive patterns and exhibit superior generalization capabilities, recent studies [7, 41] have found that such slow-thinking models exhibit low efficiency in token utilization and demonstrate overthinking behavior when addressing simple problems. Example in Figure 1 illustrates the challenges faced by both structured and unstructured CoTs. Therefore, we propose two clarifications: **1) Different types of problems may require distinct reasoning capabilities; 2) The complexity of reasoning should align with the difficulty of the problem.**

To dynamically generate appropriate reasoning structures for problems with diverse complexity, we introduce a novel paradigm of **Self-structured Chain-of-Thought (SCoT)**, which decomposes reasoning processes into minimal semantic atomic steps. To activate the model’s self-structured reasoning abilities in multimodal tasks, we further develop a full-process slow-thinking framework called

\*These authors contributed equally to this work.

†Corresponding author. Email: [xdliang328@gmail.com](mailto:xdliang328@gmail.com)

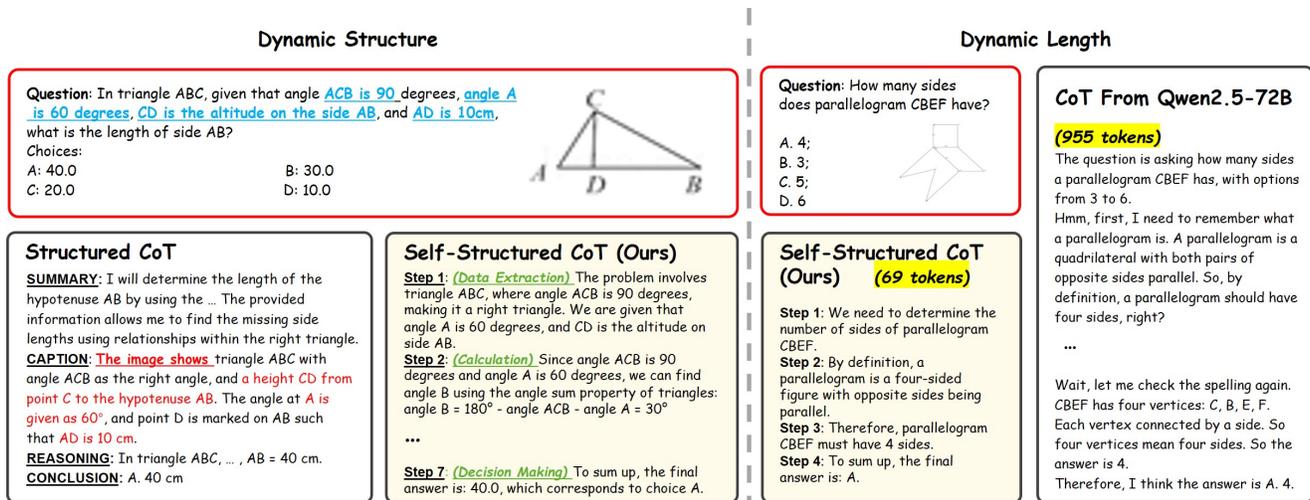


Figure 1. Comparison with structured and unstructured reasoning models. We are capable of autonomously generating dynamic structures and lengths based on the type of problem. For text-dominant questions as shown on the left, we bypass image caption and directly extracted information from the question stem. For the low-difficulty problem on the right, we use fewer tokens compared to o1-like model.

**AtomThink.** It comprises four key components, including a data engine and methods for atomic fine-tuning, policy search and atomic capability evaluation. To begin with, a data annotation engine with novel prompting and bad-case filtering strategies is used to create a novel multimodal long CoT dataset. We propose a dataset called AMATH, including 20k high-level mathematical problems with 124k atomic-granularity step annotations. Secondly, our atomic step finetuning strategy applies step-level masking to the training set, forcing our models to learn individual inference steps. During the inference phase, the model is not only capable of spontaneously generating CoT in quick mode, but also can be continuously improved with process supervision models and step search mechanisms. Lastly, we propose an atomic capability evaluation metric based on reasoning behavior clustering and step utilization calculation to evaluate the model’s performance in utilizing individual atomic steps for reasoning.

To validate the effectiveness of our method, we conduct extensive experiments on public datasets. We improve the accuracy of baseline on MathVista, MathVerse and Math-Vision by **10.9%**, **10.2%** and **7.2%**, respectively. Furthermore, our model achieves data utilization at **500%** of LLaVA-CoT while delivering superior performance, and improves inference efficiency by more than **80%**. Aiming to advance the development of multimodal high-level reasoning, we also share extended analysis in diverse reasoning ability required by visual understanding models.

In summary, our primary contributions are as follows:

- We introduce the Self-structured Chain-of-Thought to decompose the reasoning process into atomic steps. It eliminates the need for constructing structured thought tem-

plates and achieves significant improvements in both data utilization and inference efficiency.

- A comprehensive framework including modules for data annotation, atomic fine-tuning, multi-turn inference and capability evaluation, is designed to improve the reasoning ability of MLLMs.
- We validated the effectiveness of our approach on three high-level reasoning benchmarks and across various scales of MLLMs. Additionally, we present an analysis of the distribution of comprehension capabilities in MLLMs.

## 2. Related Work

### Chain of Thought in Multimodal Reasoning Tasks

Complex reasoning tasks such as mathematical computation have long been challenging for MLLMs [21, 46]. Some prior work has approached this issue from the perspective of prompt engineering, encouraging models to generate CoT, which is widely believed to enhance reasoning abilities [40, 42]. They carefully modify the input distribution to generate unstructured reasoning path without finetuning parameters. Recently, OpenAI o1 and DeepSeek R1 have demonstrated the scalability of unstructured CoT by guiding models to autonomously learn reasoning patterns through Reinforcement Learning. However, reasoning models still suffer from issues of overthinking and excessive computational consumption. Other studies have guided multimodal models to generate structured CoT by providing manually designed templates [36, 43]. Although these models incorporate visual semantic information into the reasoning process, their fixed steps constrain the diversity of reasoning actions, limiting their generalization ability on com-

plex problems.

**Long CoT Annotation for Multimodal Data** The introduction of slow thinking relies heavily on the availability of high-quality step-level annotations. Lightman et al. [17] 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 CoTs. Techniques like Quiet-STaR [48] have demonstrated how self-generated reasoning can enhance model performance without requiring manual labels. Moreover, some methods based on Monte Carlo estimation have automated the process of data collection, but they also introduce additional computational cost [26, 39]. In multimodal domain, MAVIS [50], a dataset consisting of 834k visual math problems annotated with short CoT, has been proposed. Other studies have distilled reasoning processes from short answers [51]. However, these machine-generated annotations are often too brief and challenging to segment semantically.

### 3. Method

We present the details of AtomThink for promoting MLLMs in reasoning with self-structured CoT in this section. As shown in Figure 2, AtomThink consists of four key components, including a self-structured reasoning mechanism (Sec. 3.1), a data engine (Sec. 3.2), a atomic step fine-tuning process (Sec. 3.3) and an atomic capability evaluation (Sec. 3.4).

#### 3.1. Self-structured Chain-of-Thought

To enable MLLMs to adaptively generate diverse reasoning paths in response to varying problems, akin to human cognition, we have proposed an inference method based on Self-structured Chain-of-Thought (SCoT). In contrast to structured methodologies, our approach does not constrain the model to a fixed template of thought or a predetermined sequence of reasoning steps. Instead, it empowers the model to autonomously seek out the most appropriate reasoning behaviors during the inference process.

**Multi-round Atomic Step Generation** We commence by defining the minimal predictive action with semantic consistency as an **Atomic Step**, which may constitute a single sentence or a combination thereof. Utilizing atomic steps as fundamental building blocks, we propose a multi-round prediction method to iteratively self-generate thought chains with dynamic structures. During the reasoning process, we prompt the model to predict only one minimal atomic step at a time, thereby focusing on the quality of each atomic step. Subsequently, the current prediction is

appended to the historical reasoning steps and provided as contextual input for the next prediction cycle. Our reasoning template with SCoT is shown in the Appendix Figure 9.

Due to the constraints of model size, we find that current MLLMs exhibit anomalies like reasoning stagnation and generating hallucinated content. Therefore, we use the following methods for anomaly detection and thought restart:

- **Rule-based Filter.** We employ Jaccard similarity to quantify intra- and inter-step semantic repetition, thereby mitigating thought blockages and looping phenomena. Additionally, we define a *max\_length* parameter to control the maximum length of atomic steps.
- **Temperature Accumulation.** Upon detection of an anomaly, we will perform a single-step inference anew to replace the erroneous atomic step. To enhance diversity of outcomes, we incrementally increase the temperature with each error to simulate the divergent thinking characteristic of human cognition.

**Policy Search with Process Reward Model** Given that the model spontaneously segments atomic steps during reasoning, a natural consideration is the introduction of a process reward model (PRM) to further expand the search space for predictive actions. As there are many search strategies to generate candidate actions, we categorize the existing strategies into path-wise searching and step-wise searching:

- **Path-wise Search.** 1) Majority voting combines multiple reasoning paths by selecting the most frequent outcome across them. 2) Best-of-N generates  $C$  candidate rollouts simultaneously and selects the solution with the highest score. The score can be calculated by aggregating the overall value of the entire path.
- **Step-wise Search.** 1) Greedy algorithm focuses on making the locally optimal choice at each step of the reasoning process. 2) Beam search explores multiple branches at each step 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.

In our principal experiments, we employed a step-wise beam search to extend the test-time. The Appendix A.2 and Table 9 provides a detailed description and comparative experiments of different policy search methods.

#### 3.2. Data 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 multimodal dataset, dubbed AMATH. Specifically, our data

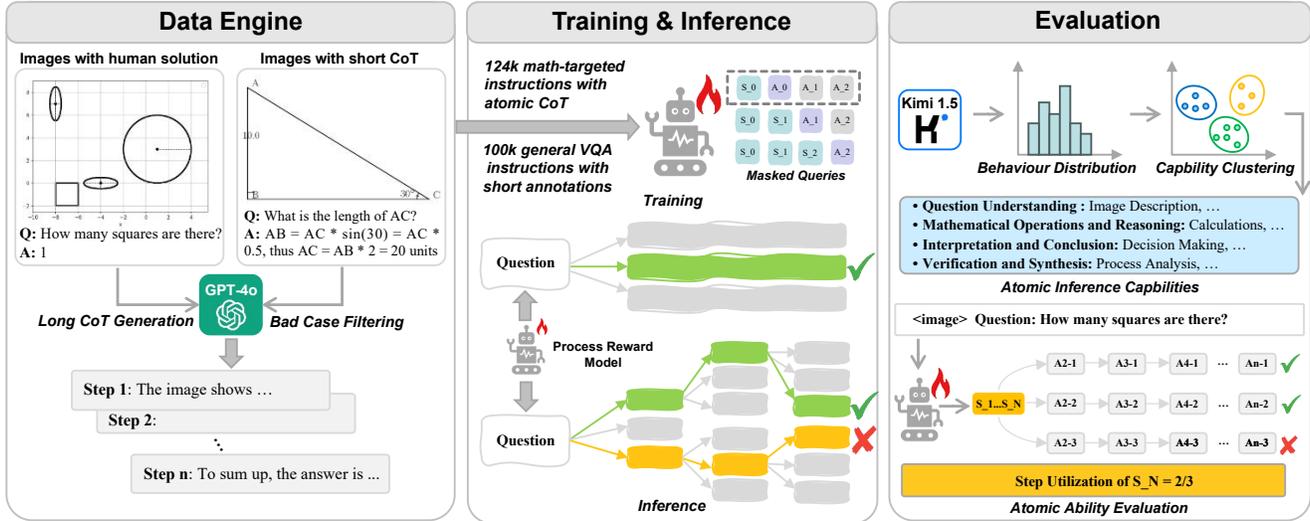


Figure 2. The overview of AtomThink framework. We annotate and filter the open-source data with long CoT to generate atomic steps for fine-tuning and PRM training. During inference, step-wise or path-wise searching strategies can be applied to find optimal policies. Finally, the behavior distribution of GPT-4o is obtained through clustering with Kimi1.5, and an outcome-based method is employed for atomic step utilization evaluation.

Source	AMATH-Metadata	AMATH-SFT
CLEVR	2056	11.9K
Geometry3K	1224	9.3K
MAVIS	1685	11.4K
TabMWP	2643	16.3K
GeomVerse	1347	9.9K
MathV360K	5632	31.6K
GeoQA+	2222	15.5K
IconQA	3199	18.1K
Total	20008	124K

Table 1. Data composition of our AMATH. 20K VQA samples are applied to generate 124K SFT data with intermediate atomic steps.

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

Table 2. Comparison of different data styles. AMATH achieves the highest GPT-4o preference score and generates longer content than vanilla CoT.

engine introduces a dynamic prompting strategy and short CoT augmentation strategy to produce multi-step reasoning paths. Subsequently, we propose a difficulty scoring mechanism coupled with a secondary review strategy to sift through and filter out erroneous instances.

**Multimodal CoT Generation.** For long CoT generation, we propose two prompt-based methods:

- **Dynamic Prompting.** Inspired by recent research [15], we propose a dynamic prompt strategy for generating atomic inference steps. Specifically, our strategy drives a LLM 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 the LLM itself. The prompt is shown in the Appendix.
- **Short CoT Augmentation.** To fully leverage existing short CoT annotations of VQA datasets, we also employ an MLLM to atomize and augment these annotations. 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.

**Bad Case Filtering.** Due to the prevalence of substantial noise within the publicly available datasets, we first employ a difficulty scoring system to filter the questions. Subsequently, a LLM is used for a secondary review to eliminate erroneous CoTs.

- **Difficulty Scoring.** To quantify the difficulty of questions, we employ Qwen2-VL-7B to sample  $N$  candidates for each question, using the win rate of  $N$  candidates as difficulty level of the question ( $N = 10$  in our paper). To enhance the efficiency of training, we have removed most questions with a difficulty level of 0.

- **Secondary Review.** Upon the generation of CoT, we utilize GPT-4o to conduct secondary review, with a particular focus on the accuracy of atomic steps and the correctness of final answers. Furthermore, we engage two professional annotators to perform a sampling inspection of our dataset.

**AMATH Dataset.** We sample multimodal reasoning data from CLEVR [13], Geometry3K [23], MAVIS [50], TabMWP [22], GeoVerse [14], Mathv360k [34], GeoQA+ [5] and IconQA [24]. For GeoVerse and MAVIS, we conduct short CoT augmentation, while the rest are generated by dynamic prompts to produce multi-step reasoning. Table 1 illustrates the distribution of our data. In Table 2, we also evaluate the quality in a subset of 500 AMATH samples with GPT-4o scoring. We have illustrated the generation and filtration examples of our dataset in Fig. 3.

### 3.3. Atomic Step Fine-Tuning

To fully exploit MLLMs for addressing multi-modal mathematical problems, we conduct fine-tuning with atomic step-wise reasoning. We have dissected CoTs from the metadata of AMATH into atomic steps and subsequently employed serialized masking to incrementally incorporate these into the historical reasoning steps, thereby generating multiple training samples (denoted as AMATH-SFT) for supervised instruction fine-tuning.

### 3.4. Atomic Capability Evaluation

Similar to human problem-solving processes, a SCoT may involve multiple reasoning abilities. However, traditional CoT methods do not focus on the ability to follow individual reasoning step 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 reasoning.

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 construct a canonical set of capabilities. As shown in Figure 6, we collect the behavior distribution of GPT-4o on AMATH dataset and use Kimi-1.5 to perform clustering, yielding clusters that each of them represents a certain ability utilized by high-level intelligent models in solving mathematical problems. We consider each cluster as a set and let  $Set(a)$  denote the cluster of an ability  $a$ .

We initially posit that models with superior atomic reasoning capabilities are more adept at leveraging recent contextual steps to further excavate answers. Hence, we can quantify a certain reasoning ability of a model based on its average probability of reaching a correct answer with its rollouts sampled from the corresponding ability set. Specifically, assume a question has  $n$  historical reasoning steps

$S = \{s_i | i = 1, \dots, n\}$ . We define the step utilization rate  $u(S)$  as the probability of reaching an answer by continuing to reason based on  $S$  averaging on  $M$  sampled rollouts:

$$u(S) = \frac{\sum_{m=1}^M \mathbb{1}[r_m \text{ is correct}]}{M}, \quad (1)$$

where  $r_m$  is the  $m$ -th rollout. Subsequently, we calculate the utilization rates of different historical steps and map the corresponding  $S$  back to the set of atomic capabilities. We compute the average utilization rate for each category in the ability set to represent the model’s atomic reasoning capability, which can be represented as follows,

$$Score(a) = \frac{1}{|Set(a)|} \sum_{S_k \in Set(a)} u(S_k). \quad (2)$$

In our experiments, we selected 160 samples from an out-of-distribution mathematical dataset (R1V-Stratos [47]), to construct a test set for atomic capability evaluation.

## 4. Experiment

### 4.1. Setup

**Baselines.** Our experiments utilize two open-source MLLMs, including LLaVA1.5-7B [20] and Llama3.2-11B-Vision [27]. With a subset of 100K multimodal question-answer pairs sampled from LLaVA-665K [20], we post-training full parameters of their language models, projectors and vision encoder as baselines. For our AtomThink models, the AMATH-SFT dataset introduced in Section 3.2, is incorporated to introduce atomic reasoning capabilities. We use a learning rate of  $2e-6$  and a batch size of 128 to fine-tune for one epoch. We select 12 cutting-edge MLLMs for comparison, including Claude 3.5 Sonnet [1], OpenAI’s o1 [30], 4o [28], 4v [29], as well as LLava-NeXT-34B [19], InternLM-XComposer2 [50], Qwen-VL-Plus [2], LLaVA-1.5-13B [20], GLLaVA-7B [10], MAVIS-7B [50], LlamaV-o1-11B [36] and LLaVA-CoT-11B [43].

**Evaluation Setting.** We evaluated the performance of our method on MathVista [25], a publicly available benchmark encompassing both general-targeted and mathematics-targeted (MathVista-M) domains. Additionally, MathVerse [52] is introduced to assess the model’s sensitivity to mathematical graphs. MathVision [38], a benchmark encompassing a diverse range of mathematical problem complexities, is also incorporated into the experiments to specifically evaluate the dynamic variations in our atomic steps. We also introduced Humanity’s Last Exam (HLE) [31], one of the most challenging benchmark, to assess the model’s reasoning capabilities under extremely difficult conditions.

Our evaluations include four inference settings, including **Direct**, **CoT**, **SCoT**, and **SCoT w/ PRM**. In the **Direct**

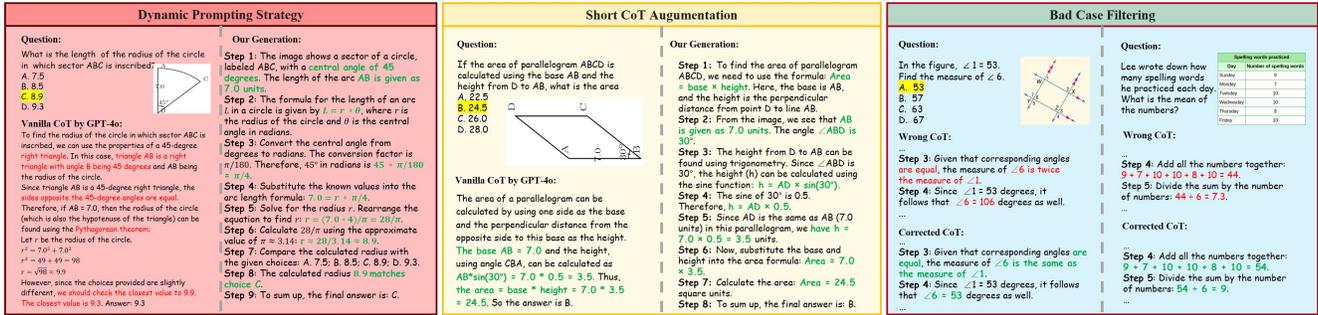


Figure 3. A case study of our data engine to generate high quality CoT. Red and green characters denote incorrect and correct responses, respectively. Compared with vanilla CoT generated by GPT-4o, our dynamic prompting strategy exhibits fewer hallucinations in every atomic steps. Utilizing existing short annotations, we can augment longer paths that encompass more details. Additionally, bad case filtering is applied to inspect low-quality noisy data within the automated pipeline.

Model	Inference	MathVista-M	MathVista	MathVerse	MathVision	HLE
Random Choice	-	-	17.9	12.4	7.2	-
Human	-	-	70.9	-	68.8	-
OpenAI o1	CoT	-	73.9	-	-	8.8
Claude 3.5 Sonnet	CoT	-	67.7	-	38.0	4.8
GPT-4o	CoT	-	63.8	-	-	3.1
GPT-4V	CoT	-	49.9	54.4	24.0	-
LLaVA-NeXT-34B	Direct	-	46.5	23.8	-	-
InternLM-XComposer2	Direct	-	57.6	16.5	14.5	-
Qwen-VL-Plus	Direct	-	43.3	11.8	10.7	-
LLaVA-1.5-13B	Direct	-	27.6	15.6	11.2	-
G-LLaVA-7B	Direct	-	53.4	16.6	-	-
MAVIS-7B	Direct	-	29.1	27.5	19.2	-
LLaMAV-o1-11B	CoT	-	54.4	-	-	-
LLaVA-CoT-11B	CoT	-	54.8	-	-	-
LLaVA1.5-7B*	Direct	23.3	27.9	10.0	9.3	4.2
AtomThink-LLaVA	SCoT	26.5 (+3.2)	29.2 (+1.3)	14.4 (+4.4)	12.7 (+3.4)	5.7 (+1.5)
AtomThink-LLaVA	SCoT w/ PRM	31.1 (+7.8)	32.1 (+4.2)	14.6 (+4.6)	12.3 (+3.0)	4.4 (+0.2)
Llama3.2-Vision-11B*	Direct	44.1	47.5	23.3	13.8	4.0
AtomThink-LlamaV	SCoT	56.9 (+12.8)	57.1 (+9.6)	31.5 (+8.2)	18.2 (+4.4)	5.4 (+1.4)
AtomThink-LlamaV	SCoT w/ PRM	59.1 (+15.0)	58.4 (+10.9)	33.5 (+10.2)	21.0 (+7.2)	4.5 (+0.5)

Table 3. Comparison of accuracy with state-of-the-art models on four benchmarks. Our AtomThink achieves consistent improvement across models of varying scales and surpasses baselines on all four benchmarks. Specially, AtomThink-LlamaV, with 11B parameters, surpasses GPT-4V by 8.5% on MathVista. The baseline models (\*) are post-trained by LLaVA100K VQA.

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 Imms-eval [16, 49]. Our AtomThink-models support two additional settings: **SCoT** and **SCoT w/ PRM**. In SCoT, our models follow a single, atomic reasoning path based purely on their learned policies, without employing any supplementary search strategies. In SCoT w/ PRM, enhanced by Qwen2.5-Math-PRM-

7B [53], we utilize step-wise beam search with a window of 3 and candidate number of 2. During the search process, the temperature for each step is initialized at 0 and incremented by 0.5 with each candidate sampling to enhance diversity.

### 4.2. Main Results

In Table 3, our AtomThink framework is applied to train LLaVA1.5-8B and Llama3.2-Vision-11B, yielding consistent performance improvements over the original models.

Method	LLaVA-CoT	AtomThink-LlamaV	AtomThink w/ PRM
Accuracy	54.8	57.1 (+2.3)	58.4 (+3.6)
Dataset Scale	100k	20k (-80%)	20k (-80%)
Tokens	1322.2	161.5 (-87.8%)	734.7 (-44.4%)
Inference Time	57.2	8.4 (-85.3%)	38.1 (-33.4%)

Table 4. Comparison with LLaVA-CoT. We not only improve inference accuracy by 3.6%, but also decrease the data and test-time resource requirement.

Samples	Accuracy
0	9.28
10k	9.67
30k	9.33
60k	11.33
90k	8.97
124k	12.45

Table 5. AtomThink-LLaVA performance improvement of MathVision-mini with dataset scaling.

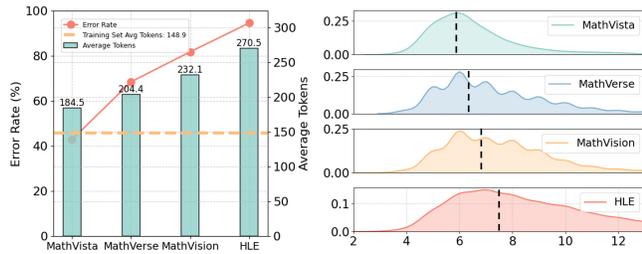


Figure 4. Comparison of the average response length in AtomThink-LlamaV over benchmarks with different complexity. (a) As tasks become more challenging, the model proactively utilizes more tokens. (b) The proportion of longer CoT containing a greater number of atomic steps increases in outputs.

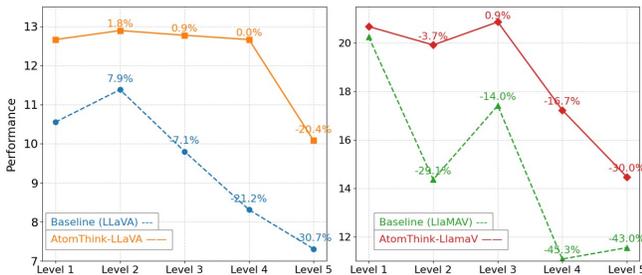


Figure 5. MathVision-mini accuracy in diverse difficulty level subsets. A higher level signifies increased difficulty. The performance decline margin of AtomThink modes are more narrow (-20.4% v.s. -30.7% in LLaVA1.5, -30% v.s. -43.0% in LlamaV).

With Self-structured CoT, the accuracy of AtomThink-LLaVA can be enhanced by 4.4% and 3.4% in MathVerse and MathVision, respectively. In a larger vision understanding model, AtomThink-LlamaV gains a higher improvement by 9.6% and 8.2%. When combined with step-wise beam search and process reward model, AtomThink-LlamaV achieves a new state-of-the-art on MathVista, surpassing GPT-4V and narrowing the gap between MLLMs and human performance. Furthermore, in the more chal-

Candidate	Ouput Tokens	Accuracy
0	2.3	13.9
1	231.9	18
2	518.6	18.3
3	822.3	23.3

Table 6. AtomThink-LlamaV performance improvement of MathVision-mini with test-time scaling. We employ Best-of-N and PRM to select the optimal step among N candidates.

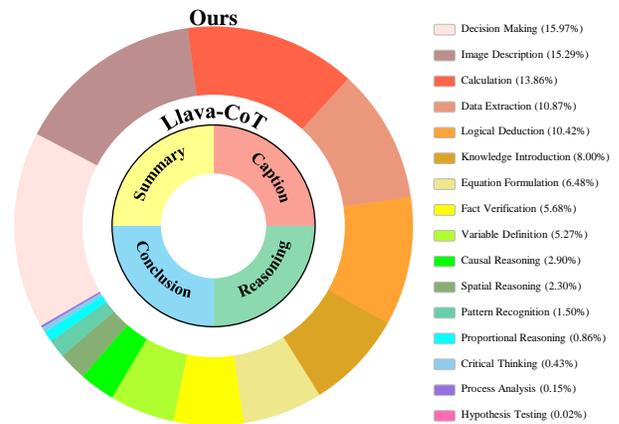


Figure 6. Atomic step distribution of AtomThink-LlamaV in R1V-Stratos-160 testset. Compared with the structured method, our model is capable of performing a variety of reasoning behaviors.

lenging and complex HLE task, despite being unable to answer the majority of questions, our two models still achieved modest improvements (1.5% and 1.4%). Overall, we increase the average performance of 7B and 11B models by 4% and 8.8%, respectively. These results demonstrate the framework’s strong generalization capability and practical usability.

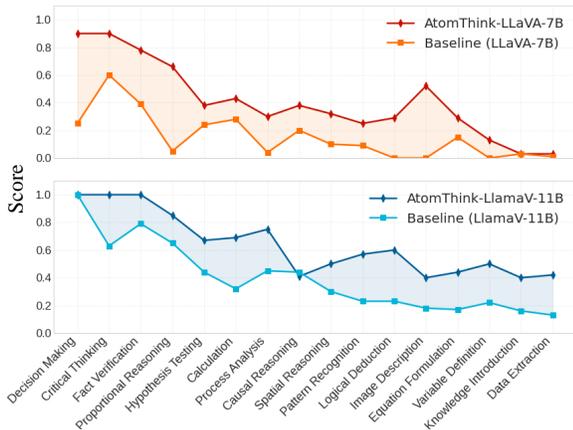


Figure 7. The evaluation of utilization efficiency across different atomic capabilities. Lower scores in initial stages of CoT (e.g. Data Extraction) indicate the presence of error accumulation.

### 4.3. Scaling Reasoning According to Difficulty

To assess the variation in the length of unstructured CoT, we present the output distribution of AtomThink-LlamaV across four benchmarks in Fig. 4. The ascending error rates indicate a sequential increase in benchmark difficulty. In subplot (a), without human intervention, the model employs a greater number of atomic steps to address more complex problems. Surprisingly, despite the average token count of AMATH-SFT being only 148.9, the outputs of our model on different test sets are progressively increasing. This suggests that the model is not merely fitting the training data but is instead exhibiting an emergent ability to autonomously explore the depth of reasoning, potentially representing the “Aha Moment” we anticipate in visual reasoning models. Moreover, although accuracy decreases with increasing difficulty level, Fig. 5 demonstrates that the decline margin is reduced when AtomThink is applied.

### 4.4. Autonomous Generation of Diverse Structures

We cluster the reasoning behaviors of GPT-4o into 16 categories and collect the distribution of atomic steps produced by AtomThink on the Stratos160 test set. The results in Fig. 6 demonstrate that, compared to structured output (LLaVA-CoT), our SCoT exhibits a more diverse range of reasoning structures. Among all categories, the higher proportion of Image Description (15.29%) and Data Extraction (10.87%) underscores the importance of perceptual capabilities. With the enhanced visual understanding abilities, the model also displays specific behaviors such as Causal Reasoning (2.9%) and Spatial Reasoning (2.3%).

### 4.5. Data Utilization and Reasoning Efficiency

In Table 4, we present a comprehensive comparison with state-of-the-art methods in terms of accuracy, dataset scale, output token count, and inference time. By utilizing only one-fifth of VQA samples, we achieve a 3.6% improvement on MathVista. Furthermore, due to our ability to provide concise responses to simpler questions, we reduce inference time by 85.3% and 33.4% per sample when not using search and employing PRM for strategic search, respectively.

### 4.6. Scaling Law in Data and Test-time

Previous research has found that scaling up data and test-time computations can enhance the reasoning in language models. Our result also discovers that this scaling law persists in multimodal models. Fig. 5 shows that increasing data scale generally promotes performance. By employing a step-wise Best-of-N strategy, we linearly increased reasoning time, with each additional candidate improving accuracy by an average of 3.1%.

### 4.7. Further Analysis

**What Kind of Capabilities Do MLLM Need in Reasoning?** Building upon the set of atomic capabilities illustrated in Fig. 6, we calculated our model’s utilization rate for each category of steps using Eq. (2). Results in Fig. 7 reveal that as the given historical steps approach the beginning of the reasoning chain (e.g. Image Description and Data Extraction), prediction error rate continuously increases. This error accumulation effect prompts us to focus on the quality of reasoning in initial stages. In future work, we can mitigate the rate of error accumulation by adjusting data ratios and designing sampling strategies.

**What Kind of Information Do PRM Focus on?** In Table 3, we find that even the reasoning process heavily relies on visual dominant inputs, the highest performance is achieved by using a language PRM. Meanwhile, we have tried to train a vision-language PRM of text and visual modalities data. Unfortunately, the results indicate that the language reward model achieves better performance than the MLLM. This suggests that the current PRM paradigm may not be adept at utilizing visual information. Exploring how to leverage multimodal features to correct the reasoning process will be a direction we need to investigate.

## 5. Conclusion

To mitigate overthinking and structured output, we have proposed a self-structured chain of thought method. It ensures reasoning efficiency while adaptively generating a variety of atomic steps. Subsequently, we introduced AtomThink, a comprehensive deep reasoning framework that encompasses data engineering, model training, inference, and evaluation. The experimental results demon-

strate that our method consistently enhances the model’s diverse behaviors during the problem-solving process, leading to improved reasoning performance across various multimodal benchmarks. This work paves the way for developing generalized slow-thinking models and provides novel insights for understanding multi-modal reasoning patterns.

## References

- [1] Anthropic. Claude 3.5 sonnet. 5
- [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. 5
- [3] Li Bo. open-r1-multimodal, 2025. 1
- [4] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13817–13827, 2024. 2
- [5] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P Xing, and Liang Lin. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv preprint arXiv:2105.14517*, 2021. 5
- [6] 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. 2
- [7] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024. 1
- [8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 2
- [9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2
- [10] 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. 5
- [11] 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
- [12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1
- [13] 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. 5
- [14] 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*. 5
- [15] Benjamin Klieger. g1: Using llama-3.1 70b on groq to create o1-like reasoning chains, 2024. 4
- [16] 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
- [17] 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*. 3
- [18] 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
- [19] 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. 5
- [20] Haotian Liu, Chunyuan Li, Qingyuan Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 5, 2
- [21] 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
- [22] 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*. 5
- [23] 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. 5
- [24] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021. 5
- [25] 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. 5
- [26] 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. 3
- [27] Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. 5
- [28] OpenAI. Gpt-4o system card, . 5
- [29] OpenAI. Gpt-4v(ision) system card, . 5
- [30] OpenAI. Openai o1 system card, . 1, 5, 2
- [31] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, et al. Humanity’s last exam. *arXiv preprint arXiv:2501.14249*, 2025. 5
- [32] 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
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2
- [34] 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. 5
- [35] 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. 1
- [36] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. Llamav-01: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*, 2025. 1, 2, 5
- [37] 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, 2
- [38] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 5
- [39] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Mathshepherd: Verify and reinforce llms step-by-step without human annotations. In *ACL*, pages 9426–9439, 2024. 3, 2
- [40] 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
- [41] Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025. 1
- [42] 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, 6
- [43] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. 1, 2, 5
- [44] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024. 2
- [45] 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
- [46] 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
- [47] Ya-Qi Yu, Minghui Liao, Jihao Wu, and Chao Weng. R1-vision: Let’s first take a look at the image. <https://github.com/yuyq96/R1-Vision>, 2025. Accessed: 2025-02-08. 5, 1
- [48] 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. 3
- [49] 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
- [50] 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. 3, 5
- [51] 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. 3
- [52] 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
- [53] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025. 6, 1

# Can Atomic Step Decomposition Enhance the Self-structured Reasoning of Multimodal Large Models?

## Supplementary Material

### A. Implementation Details

#### A.1. Policy Models

In this section, we provide more implementation details for baseline models and our framework. We post-train them using AMATH-SFT and a sub-sampled dataset of LLaVA665K, containing 100k samples. During this process, the weights of LLM, projector and vision encoder are fully fine-tuned. Specifically, we utilize the Llama-factory framework to train the models and the hyperparameters are listed in Table 7.

Parameter	LLaVA1.5-7B	Llama3.2-V-11B
Learning Rate	2e-6	2e-6
Epochs	1	1
Batch Size	128	128
Context Length	4096	4096
Seed	42	42
Precision	FP16	BF16
GPU	32 NVIDIA V100	8 NVIDIA A800
FSDP	True	True
DeepSpeed	Zero3	Zero3

Table 7. Comparison of Parameters for post-training LLaVA1.5-7B and Llama-3.2-Vision-11B.

#### A.2. Atomic Search with PRM.

With the fine-tuned MLLM capable of atomic step reasoning, we apply well-trained PRM, Qwen2.5-Math-PRM-7B [53], for providing feedback. 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. Unlike traditional token-based search strategies, we sample candidates using atomic steps as the fundamental unit.

**Path-wise Search.** In path-wise searching, we build upon prior work [35, 37] 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 simul-

aneously 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.

### B. Attempts on training a R1-like MLLM

Recently, the introduction of DeepSeek-R1 [12] has demonstrated that reinforcement learning can effectively enhance autonomous reasoning capabilities. GRPO (Generalized Reward Prediction Optimization) as its main training strategy, focuses on optimizing the prediction of rewards in complex environments to improve decision-making and policy learning. Several outstanding open-source repositories have explored the application of GRPO in the visual domain [3, 47].

With the setup of R1V [47], we also attempt to use GRPO to address complex reasoning problems. To facilitate a fair comparison, we sample a 3K subset from AMATH-metadata, equivalent in scale to R1V-CLEVR, and conducted experiments on Qwen2-VL-2B. The reward function is divided into format rewards and accuracy rewards. The Figure 8 illustrates the changes in candidate length and accuracy rewards during the training process. Due to the use of more diverse and complex training data, we generate longer CoT completion length and lead to higher error

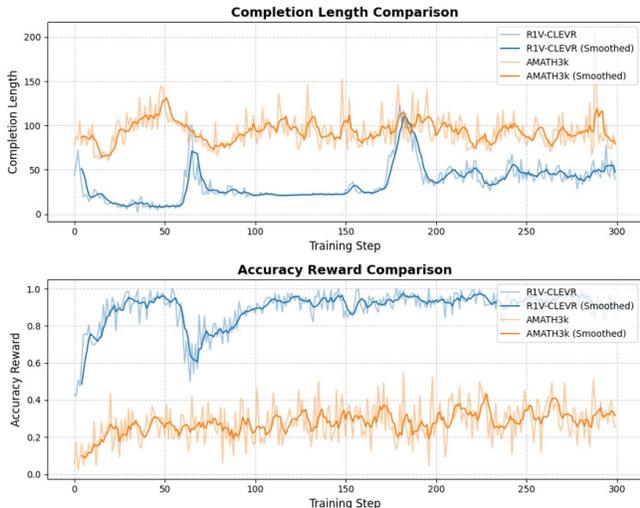


Figure 8. Comparison with a DeepSeek-R1 like framework using reinforcement learning. A 3k subset of AMATH is sampled for fair comparison.

rates. Although there is an initial improvement in accuracy rewards during the early stages of AMATH3K training, it plateaus around 30%. Additionally, neither set of experiments exhibit the CoT growth phenomenon observed in R1. We think that pure reinforcement learning methods face greater challenges in multimodal tasks, potentially relate to task difficulty, model foundational capabilities and reward settings.

### C. Prompts Design

In this section, we present the prompt used in self-structured CoT 9 and multimodal CoT annotation engine. Prompts in data engine include: long CoT generation (Figure 10), short CoT augmentation (Figure 11), data filtering (Figure 12), and quality scoring (Figure 13).

### D. Cases of AtomThink output

In Figure 15 and Figure 16, we present the SCoT outputs generated by Llama3.2-Vision-11B models trained with AtomThink. Compared to original models, we are able to produce a dynamic thinking process similar to OpenAI-o1 [30]. The models tend to use image features to decompose challenging mathematical problems into multiple atomic sub-questions, which are then solved step by step. It is capable of generating responses of varying lengths based on the difficulty of problem and exhibits diverse reasoning behaviors (including Fact Verification, Spatial Reasoning and Logical Deduction). The results demonstrate that our outputs are more accurate in recognizing visual information and reduce reasoning hallucinations.

## E. Cases of AtomMATH Dataset

To further illustrate the AMATH dataset we released, we randomly select several examples. Figure 17 and Figure 18 show two case from AMATH-SFT.

## F. Early experiments

In this section, we provide more early exploration in SFT based slow thinking model. We use LLaVA-Llama3 [20] and EMOVA-8B [6] to perform experiments.

### F.1. Implementation Details

**Basic Model.** For LLaVA-Llama3 [20], we choose the pre-trained ViT-L/14 of CLIP [33] as the vision encoder and Llama3-8B [9] as our LLM. To align visual features with the LLM, we incorporate a Multi-Layer Perceptron (MLP) as a projector between the visual encoder and the language model. For EMOVA-8B [6], we use the original setting of EMOVA that uses InternViT-6B [8] and LLaMA-3.1-8B [9]. The C-Abstractor [4] with two ResBlocks is adopted as the projector.

The training of LLaVA-Llama-3-8B follows a structured two-stage process [20]. In our experiment, we only load its weights from pre-training stage and deploy supervised fine-tuning. During SFT, the training data comprises the LLaVA-Instruct-665k, a 46k subset of PRM800k and our AMATH-SFT dataset. The weights of language model and MLP projector are unfrozen. The model undergoes an epoch of training with a reduced learning rate of  $2e-5$  and batch size of 128. To create AtomThink-EMOVA, we post-train EMOVA using AMATH-SFT and a sub-sampled dataset of EMOVA-SFT-4m, containing 200k samples. During this process, the weights of the LLM and the C-Abstractor projector are updated. EMOVA is fine-tuned for 1 epoch with a batch size of 128 and a learning rate of  $2e-6$ .

**PRM Setting.** We initially fine-tuned a large language model to introduce textual process supervision. We utilize the pre-trained Qwen2.5-Math-7B-Instruct [44] and Math-psa-7B [37] models as our foundational architectures. Qwen2.5-Math-7B-Instruct is an open-source large language model specifically designed for mathematical reasoning tasks. Math-psa-7B is a text-based process supervision model trained using datasets such as PRM800K [18], Math-Shepherd [39] and MATH-APS [37]. Low-Rank Adaptation (LoRA) is applied to fine-tune with the following parameters: rank ( $r$ ) of 8, alpha scaling factor of 32, dropout rate of 0.1, and targeting the  $q$  and  $v$  projectors. Training is conducted over one epoch with a batch size of 256 and a learning rate of  $1e-5$ . We sample a 20k-instance training set from PRM800K and combine it with the AMATH-PRM dataset, which is derived from multimodal CoT annotations,

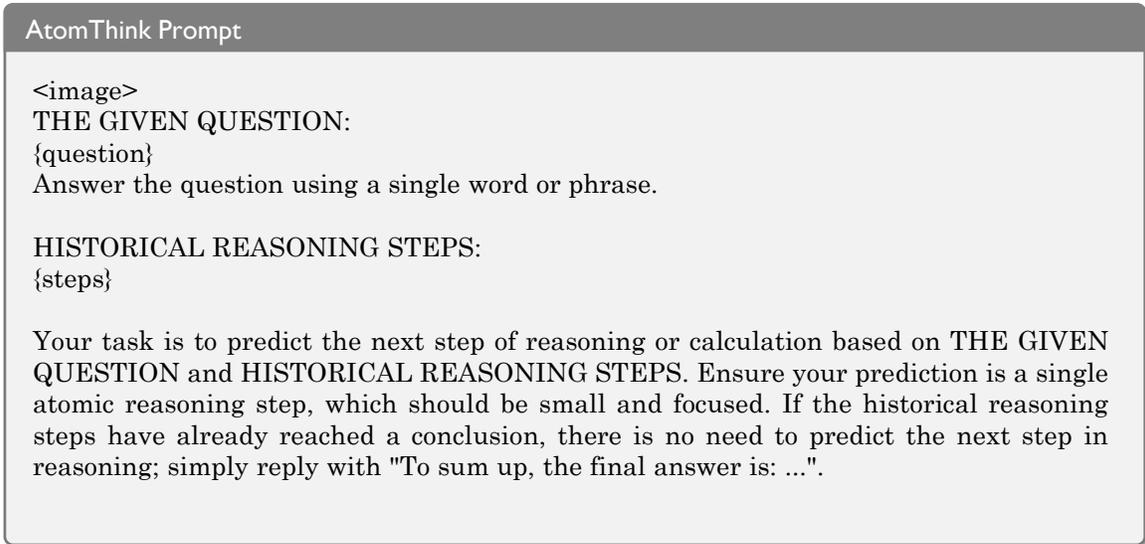


Figure 9. AtomThink template for generating Self-structured CoT. The model takes an image and a question as input, generating an atomic step at each iteration. These steps are then concatenated into the historical reasoning steps, which are fed into model for the next round of reasoning.

Model	Inference	MathVista			MathVerse					
		General	Math	Total	TL	TD	VI	VD	VO	Total
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-Llama3	Direct	34.4	27.2	30.5	16.0	19.3	16.2	13.1	15.0	15.9
AtomThink-LLaVA-Llama3	SCoT	<b>36.9</b>	<b>37.0</b>	<b>36.6</b>	<b>22.2</b>	<b>26.6</b>	<b>24.1</b>	<b>20.9</b>	<b>17.9</b>	<b>22.4</b>
AtomThink-LLaVA-Llama3	SCoT w./ PRM	<b>36.5</b>	<b>41.3</b>	<b>39.1</b>	<b>36.1</b>	<b>42.4</b>	<b>30.0</b>	<b>36.8</b>	<b>28.6</b>	<b>34.7</b>
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	SCoT	48.7	<b>54.4</b>	<b>51.8</b>	<b>36.5</b>	<b>42.4</b>	<b>34.1</b>	<b>32.9</b>	<b>29.7</b>	<b>35.1</b>
AtomThink-EMOVA	SCoT w./ PRM	48.9	<b>57.0</b>	<b>53.3</b>	<b>42.1</b>	<b>51.5</b>	<b>39.0</b>	<b>36.7</b>	<b>33.1</b>	<b>40.5</b>

Table 8. Comparison of accuracy on MathVista and MathVerse. Our AtomThink-LLaVA-Llama3 outperforms the baseline in all sub-tasks across two benchmarks, achieving an average improvement of 14.2%.

to serve as our fine-tuning data. All the samples include question, historical steps, and current step, with each current step being assigned a label of either correct or incorrect. In line with OpenR’s settings, we designate “\n\n\n\n\n” as the step separator and return the conditional probability of the current step being correct.

## F.2. Results

**Multimodal Performance Improvement.** In Table 8, 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 and narrowing the gap between MLLMs and human performance. On MathVista, it also achieves performance to 53.3%. These results demonstrate the framework’s strong generalization capability. In Figure 19, we compare AtomThink with the state-of-the-art open-source inference strategy, g1<sup>1</sup>, 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 re-

<sup>1</sup><https://github.com/bklieger-groq/g1>

## Dynamic Prompt to Generate Long CoT

<SYSTEM>

You are an expert AI assistant that explains your reasoning step by step. Your task is to continue your previous conversation and predict the next step in reasoning. Decide if you need another step or if you're ready to give the final answer. Respond in JSON format with 'content', and 'next\_action' (either 'continue' or 'final\_answer') keys.

1. Ensure your output is a single atomic reasoning step, which should be small and focused.
2. Ensure that your reasoning incorporates all relevant details from the provided image.
3. Break down your explanation into clear, concise steps. Use as many reasoning steps as possible while avoiding unnecessary or redundant information.
4. In your reasoning process, utilize various approaches to explore the answer comprehensively, ensuring a thorough analysis.
5. Base your reasoning strictly on the information available in the image and prior context to prevent inaccuracies.

Examples of valid responses:

{examples}

<USER>

{image}

{question}

<ASSISTANT>

I will now think step by step following my instructions.

<ASSISTANT>

```
```json
{
  "content": "Step 1: The image shows ...",
  "next_action": "continue"
}```
```

.....

<ASSISTANT>

```
```json
{
  "content": "Step N: The final answer is: ...",
  "next_action": "final_answer"
}```
```

Figure 10. Dynamic prompt for long CoT generation. Inspired by previous work, we designed a dynamic prompt template that generates reasoning steps for each iteration. It effectively identifies the input visual information to generate detailed image captions and fine-grained atomic steps.

## LLM Data Augmentation Prompt

You are an advanced multimodal large language model. Your task is to generate a Chain of Thought (CoT) reasoning for a question based on a provided image and a reference answer. Break down your reasoning into clear, logical steps that are easy to follow.

### Requirements:

1. Construct a logical, step-by-step thought process using information from the image and the reference answer, along with any relevant external knowledge.
2. Ensure each step builds on the previous one and leads to the final answer.
3. Make necessary **\*\*inferences\*\*** based on the image content and additional knowledge (e.g., science, mathematics, geography).
4. In the last step, provide a concise, well-supported answer to the question, concluding with "Step N: To sum up, the final answer is: xxx."

---

### Example Format:

[Input]  
{input example}

---

[Output]

.....

---

### Note: Ensure that your reasoning is **\*\*clear, logical, and complete\*\***, with no omitted steps. This will demonstrate how the answer is derived from both the image content and the reference answer.

---

[Your Input]

{image}  
{question}  
{reference answer}

[Your Output]

Figure 11. Prompt for short answer augmentation. Using the current math VQA dataset, which already includes short answers and CoTs, we apply this template to enhance and generate detail atomic steps.

Model	Method	MathVista-M	MathVista-G	MathVerse
EMOVA-200k	Direct	51.1	52.4	33.3
AtomThink	Direct	52.4	53.9	35.7
	Quick Think	54.2	46.7	38.0
w/. Path-wise	Majority Voting	48.8	49.4	39.0
	BoN-Last	51.2	46.8	41.3
	BoN-Avg	58.7	40.5	38.7
	BoN-Min	53.7	53.2	40.0
w/. Step-wise	Greedy	46.3	45.6	38.3
	Beam Search	57.1	53.2	45.3

Table 9. 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.

Candidate	AtomThink-EMOVA	AtomThink-LLaVA
1	38.0	22.5
2	37.5	23.0
3	38.5	31.5
4	41.5	33.5
5	42.5	35.5
6	45.5	36.0
10	45.5	38.0

Table 10. Test-time scaling law of MLLMs. Results suggest that increasing the number of candidates during inference can enhance models’ performance on mathematical tasks, aligning with the finding of test-time scaling laws in language models.

liance 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.

**Test-time scaling law in multimodal tasks** In this subsection, we evaluate the impact of inference-time expansion on experimental outcomes using a 200-sample subset of the MathVerse dataset. We employ the AtomThink-EMOVA-8B and AtomThink-LLaVA-8B in Sec. 4 as base models, maintaining a fixed beam size of 1, and increase the number of candidates per sampling to extend test time. The results in Table 10 indicate that, as the number of candidates increases, both models exhibit consistent performance improvements on multimodal mathematical tasks. Specifically, AtomThink-EMOVA-8B’s accuracy rises from 38.0% with a single candidate to 45.5% with 6 candidates, while AtomThink-LLaVA-8B’s accuracy improves from 22.5% to 36.0% over the same range.

**Evaluate CoT Capability with Various Prompts** Chain-of-Thought prompting has been demonstrated to enhance the performance of large language models by guiding them to deliberate on their predictions [42]. However, previ-

ous studies have indicated that such benefits emerge predominantly in models exceeding 100 billion parameters. In our investigation, we extend this analysis to multimodal large language models by employing various widely-used prompts to induce step-by-step reasoning in the 8B-parameter LLaVA-Llama3 model. In Table 11, the prompt “Answer the question using a single word or phrase.” forces model output directly and achieves the highest accuracy at 18.5%. In contrast, prompts explicitly instructing step-by-step reasoning, such as “Let’s think step by step.” results in lower accuracies of 11.1%. Other prompts that encourage CoT output also lead to a significant decrease in reasoning accuracy. These findings suggest that incorporating slow-thinking capabilities into smaller models presents substantial challenges.

## G. Limitations

Due to a lack of computational resources, we did not conduct research on larger-scale visual models. Additionally, although we performed manual sampling and screening during data creation, it may not cover all cases and atomic steps.

Prompt	MathVerse200
Answer the question using a single word or phrase.	18.5
Let's think step by step.	11.0
First perform reasoning, then finally answer the question and provide the final value at the end in the following format: Answer: xxx.	9.5
Answer the following question step by step concisely.	14.5
Given the following problem, reason and give a final answer to the problem. Your response should end with "The final answer is [answer]" where [answer] is the response to the problem.	15.5
Answer the following question. The last line of your response should be of the following format: 'Answer: xxx' where 'xxx' is the answer. Think step by step before answering.	14.0

Table 11. Evaluate the CoT performance of LLaVA-Llama3-8B using different prompts. Results indicate that smaller multimodal large language models exhibit limited CoT capabilities.

## LLM Data Filtering Prompt

### ### Instruction: Checking Answer Correctness

Given the question, image, and ground truth solution, follow these steps to determine if the provided response is correct, including both the reasoning steps and final answer.

1. Analyze the Question and Image:
  - Ensure you understand the question statement and the context provided by the image.
2. Compare the Provided Answer with Ground Truth:
  - Verify that the steps, logic, and reasoning in the provided answer align with the ground truth solution.
  - Check for calculation correctness and factual consistency at every step of the answer.
3. Evaluate the Final Result:
  - Ensure the final answer matches the ground truth both in value and format.
4. Determine Correctness:
  - If all steps, logic, calculations, and the final result match the ground truth exactly, return `True`.
  - If there is **any discrepancy** in the process, logic, or result, return `False`.

---

### ### Example Format:

[Input]  
{input example}

---

[Output]  
True or False

---

### ### Note:

Only return True or False based on the correctness evaluation. Do not provide any additional comments, explanations, or intermediate outputs.

[Your Input]

{image}  
{question}  
{ground truth}  
{response}

[Your Output]

Figure 12. Prompt for filtering wrong CoT. Due to the quality gap between the reasoning steps generated by the AI assistant and human annotations, we employ this template to double-check. It filters out samples with incorrect answers and reasoning processes.

### LLM Scoring Prompt

Please evaluate the quality of the Chain-of-Thought (CoT) reasoning provided by the AI assistant for the given question and answer. Consider the correctness of reasoning, logical coherence, readability, detailing, and depth of understanding of the problem. Additionally, evaluate whether the model effectively addresses all parts of the question and generates accurate intermediate steps leading to the final answer.

Start by explaining your judgment, highlighting strengths or weaknesses in the reasoning. After your explanation, rate the quality of the CoT reasoning on a scale from 1 to 10, where 1 means significant flaws and 10 indicates a thorough, logical, and well-structured response. Please explain your reasons for evaluation first, and then end in the format of "Rating: [[X]]", where X is your score.

[Question]  
{question}

[Answer]  
{answer}

[Start of Assistant's Chain-of-Thought]  
{cot}  
[End of Assistant's Chain-of-Thought]

---

[Your output]

Figure 13. Prompt for GPT scoring. We use this template and GPT-4o to quantitatively evaluate the quality of the generated data. The results show that our AMATH data outperforms human annotations in terms of AI preference scores.

### Atomic Ability Clustering

You are a data analysis expert proficient in extracting significant information from large datasets. Analyze the provided reasoning step data and extract the most relevant distinct reasoning capability labels.

#### ### Requirements:

1. Analyze each reasoning trajectory to identify fine-grained cognitive abilities
2. Consolidate fundamental reasoning competencies into representative labels
3. Prioritize capabilities demonstrating logical inference and problem-solving

#### ### Example Format:

[Input]

{Reasoning Data}

[Output]

1. Graph Analysis
2. Variable Definition
3. Problem Reformulation
4. ....

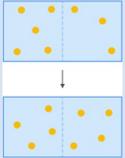
### Note: Maintain **concise, specific, and mutually exclusive** labels that comprehensively cover all demonstrated capabilities. This taxonomy will inform core reasoning competencies.

Figure 14. Prompt for clustering the reasoning behaviors in GPT-4o.

Meta Sample

**Question:** How many solute particles are present in the left compartment before reaching equilibrium?

**Answer:** The answer is 5.



AMATH-SFT

Step 1: **(Image Description)** - The image depicts two compartments separated by a dashed line, which represents a semi-permeable barrier. The top part of the image shows the initial state before equilibrium, while the bottom part shows the state after equilibrium is reached. Yellow circles represent solute particles distributed across the compartments.

Step 2: **(Data Extraction)** - In the top part of the image (before equilibrium), we focus on the left compartment to count the number of solute particles. The yellow circles in this compartment represent the solute particles.

Step 3: **(Pattern Recognition)** - By visually inspecting the left compartment in the top part of the image, we observe that there are five yellow solute particles present. This count is based on the distinct yellow circles visible in the left compartment.

Step 4: **(Fact Verification)** - The count of five solute particles in the left compartment is consistent with the visual evidence provided in the image. No additional solute particles are hidden or ambiguous in the depiction.

Step 5: **(Decision Making)** - To sum up, the final answer is: 5.

Meta Sample

**Question:** What activity is the person in the foreground engaged in?

**Choices:** A. fishing B. running C. **skateboarding** D. eating



AMATH-SFT

Step 1: **(Image Description)** - The image shows a person in the foreground performing an activity in a concrete bowl-like structure, which resembles a skate park. The individual is positioned on a board with wheels, which is a skateboard.

Step 2: **(Knowledge Introduction)** - Skateboarding is an activity commonly performed in skate parks or similar environments, where individuals use skateboards to perform tricks, maneuvers, or simply ride within the structure.

Step 3: **(Data Extraction)** - The person is wearing casual, loose-fitting clothing, including a tank top, jeans, and a cap. This attire is typical for skateboarding, as it allows for comfort and freedom of movement during the activity.

Step 4: **(Spatial Reasoning)** - The individual is in a crouched position with knees bent and arms extended outward. This posture is indicative of maintaining balance and control, which is essential for skateboarding, especially when navigating curves or performing tricks in a skate park.

Step 5: **(Logical Deduction)** - The presence of a skateboard, the skate park setting, and the individual's posture and attire all align with the activity of skateboarding. None of the other options (fishing, running, or eating) are consistent with the visual evidence or context.

Step 6: **(Decision Making)** - To sum up, the final answer is: skateboarding.

Figure 15. Case of AtomThink-LlamaV output. Based on the type of problem and information provided, model autonomously explores diverse reasoning behaviors, e.g. Fact Verification, Spatial Reasoning and Logical Deduction.

**Meta Sample**

**Question:** Chords JF and BC intersect at K. If BK=8, KC=12, and KF=16, find JK

**Choices:** A. 6 B. 12  
C. 15 D. 18

**AMATH-SFT**

Step 1: **(Variable Definition)** - Define the variables based on the given information: BK = 8, KC = 12, KF = 16, and we need to find JK.

Step 2: **(Knowledge Introduction)** - Use the intersecting chords theorem, which states that if two chords intersect, the products of the lengths of the segments of each chord are equal. Therefore,  $BK * KC = JK * KF$ .

Step 3: **(Equation Formulation)** - Formulate the equation using the intersecting chords theorem:  $BK * KC = JK * KF$ . Substitute the known values:  $8 * 12 = JK * 16$ .

Step 4: **(Calculation)** - Calculate the product of BK and KC:  $8 * 12 = 96$ . Then solve for JK by dividing both sides by 16:  $JK = 96 / 16 = 6$ .

Step 5: **(Decision Making)** - To sum up, the final answer is: A.

**Meta Sample**

**Question:** How many pineapples are there?

**Choices:** A. 5 B. 4  
C. 3 D. 2 E. 1

**AMATH-SFT**

Step 1: **(Image Description)** - The image shows three distinct pineapples arranged in a row.

Step 2: **(Fact Verification)** - By visually confirming, there are indeed three separate pineapples in the image.

Step 3: **(Decision Making)** - To sum up, the final answer is: C (3).

**Meta Sample**

**Question:** As shown in the figure, the radius of circle O is 3.0, PA and PB are tangent to circle O at points A and B respectively, OP = 6.0, then the length of PB is ()

**Choices:** A. 3  
B.  $2\sqrt{3}$   
C.  $3\sqrt{3}$   
D. 6

**AMATH-SFT**

Step 1: **(Image Description)** - The image shows a circle with center O and radius 3.0. Two tangents, PA and PB, are drawn from an external point P to the circle at points A and B, respectively. The distance from O to P is given as OP = 6.0.

Step 2: **(Knowledge Introduction)** - A key property of tangents to a circle is that tangents drawn from a common external point to the circle are equal in length. Therefore,  $PA = PB$ .

Step 3: **(Knowledge Introduction)** - Another important property of tangents is that the radius of the circle is perpendicular to the tangent at the point of tangency. Thus, angle OPA = 90° and angle OPB = 90°. This means that triangle OAP and triangle OBP are right triangles.

Step 4: **(Variable Definition)** - In the right triangle triangle OAP, the hypotenuse is OP = 6.0, one leg is the radius OA = 3.0, and the other leg is the tangent PA, which we need to calculate. Since PA = PB, finding PA will also give the length of PB.

Step 5: **(Equation Formulation)** - Using the Pythagorean theorem for triangle OAP, we have:  
 $OP^2 = OA^2 + PA^2$   
 Substituting the known values:  
 $6^2 = 3^2 + PA^2$

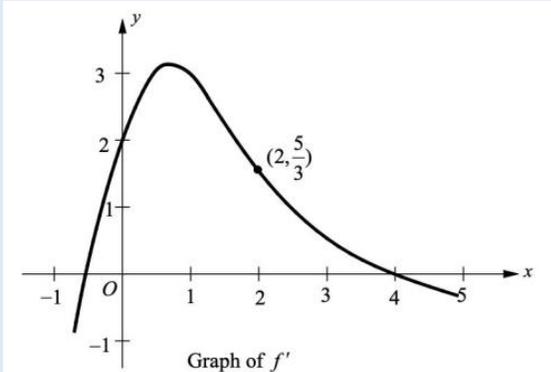
Step 6: **(Calculation)** - Simplify the equation:  
 $36 = 9 + PA^2$   
 $PA^2 = 36 - 9 = 27$   
 $PA = \sqrt{27} = 3\sqrt{3}$

Step 7: **(Logical Deduction)** - Since PA = PB, the length of PB is also  $3\sqrt{3}$ .

Step 8: **(Decision Making)** - To sum up, the final answer is: C ( $3\sqrt{3}$ ).

Figure 16. Case of AtomThink-LlamaV output. Depending on the difficulty of problem, it generates shorter or longer CoTs without human supervision.

## Meta Sample



Let  $f$  be twice differentiable function on the interval  $-1 < x < 5$  with  $f(1) = 0$  and  $f(2) = 3$ . The graph of  $f'$ , the derivative of  $f$ , is shown above. The graph of  $f'$  crosses the  $x$ -axis at  $x = -0.5$  and  $x = 4$ . Let  $h$  be the function given by  $h(x) = f(\sqrt{x+1})$ . Which is the equation for the line tangent to the graph of  $h$  at  $x = 3$ ?

(A)  $y = \frac{5x}{12} + \frac{7}{4}$  (B)  $y = \frac{5x}{12} + \frac{5}{4}$  (C)  $y = \frac{7x}{12} + \frac{7}{4}$

## AMATH-SFT

Step 1: We need to find the equation of the tangent line to the graph of the function  $h(x) = f(\sqrt{x+1})$  at  $x = 3$ . We are given that  $f$  is twice differentiable on the interval  $-1 < x < 5$ , with  $f(1) = 0$  and  $f(2) = 3$ . The graph of  $f'$ , the derivative of  $f$ , is provided, and it crosses the  $x$ -axis at  $x = -0.5$  and  $x = 4$ .

Step 2: The graph of  $f'$  shows that  $f'(2) = \frac{5}{3}$ . This information will be useful for finding the slope of the tangent line to  $h$  at  $x = 3$ .

Step 3: To find the equation of the tangent line to  $h(x)$  at  $x = 3$ , we need to determine  $h(3)$  and  $h'(3)$ . We start by finding  $h(3)$ :  $h(3) = f(\sqrt{3+1}) = f(2)$ . Given that  $f(2) = 3$ , we have  $h(3) = 3$ .

Step 4: Next, we need to find  $h'(3)$ . Using the chain rule,  $h'(x) = f'(\sqrt{x+1}) \cdot \frac{1}{2}(x+1)^{-\frac{1}{2}}$ . Evaluating this at  $x = 3$ , we get  $h'(3) = f'(2) \cdot \frac{1}{2}(4)^{-\frac{1}{2}}$ .

Step 5: We know from the graph that  $f'(2) = \frac{5}{3}$ . Therefore,  $h'(3) = \frac{5}{3} \times \frac{1}{2}(4)^{-\frac{1}{2}} = \frac{5}{3} \times \frac{1}{2} \times \frac{1}{2} = \frac{5}{3} \times \frac{1}{4} = \frac{5}{12}$ .

Step 6: The slope of the tangent line to  $h$  at  $x = 3$  is  $\frac{5}{12}$ , and the point of tangency is  $(3, 3)$ . Using the point-slope form of the equation of a line,  $y - 3 = \frac{5}{12}(x - 3)$ , we can rewrite this as  $y = \frac{5}{12}x + 3 - \frac{5}{12} \cdot 3 = \frac{5}{12}x + 3 - \frac{5}{4} = \frac{5}{12}x + \frac{7}{4}$ . Therefore, the equation of the tangent line is  $y = \frac{5}{12}x + \frac{7}{4}$ .

Step 7: To sum up, the final answer is: A.

Figure 17. Case of AMATH-SFT.

## Meta Sample

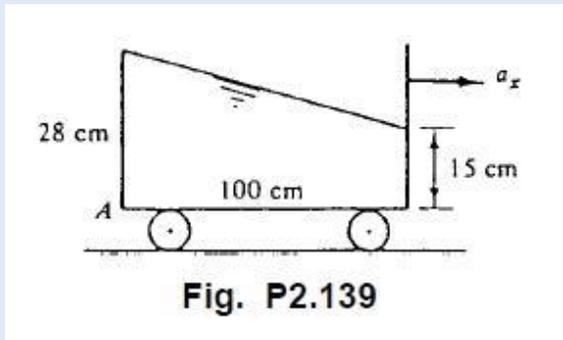


Fig. P2.139

The tank of liquid in the figure P2.139 accelerates to the right with the fluid in rigid-body motion. Compute  $a_x$  in  $m/s^2$ .

(A)  $1.00 m/s^2$  (B)  $1.50 m/s^2$  (C)  $1.28 m/s^2$

## AMATH-SFT

Step 1: The problem involves determining the acceleration of a tank of liquid in rigid-body motion based on the given dimensions and the fluid surface slope.

Step 2: The image shows a tank accelerating to the right with a liquid inside. The liquid surface is inclined, with the left side at 28 cm and the right side at 15 cm. The horizontal distance between these points is 100 cm.

Step 3: To solve this problem, we need to use the relationship between the slope of the liquid surface and the acceleration of the tank. The slope of the liquid surface is given by the difference in height divided by the horizontal distance.

Step 4: The slope of the liquid surface in a tank accelerating horizontally can be described by the equation:  $slope = \frac{(h_1 - h_2)}{L} = \frac{a_x}{g}$ , where  $h_1$  and  $h_2$  are the heights of the liquid at two points,  $L$  is the horizontal distance between these points,  $a_x$  is the horizontal acceleration, and  $g$  is the acceleration due to gravity ( $9.81 m/s^2$ ).

Step 5: Calculate the slope using the given heights and horizontal distance:  $slope = \frac{(28\text{ cm} - 15\text{ cm})}{100\text{ cm}} = \frac{13\text{ cm}}{100\text{ cm}} = 0.13$ .

Step 6: Using the slope and the equation  $slope = \frac{a_x}{g}$ , solve for  $a_x$ :  $a_x = slope \cdot g = 0.13 \times 9.81 m/s^2 = 1.2753 m/s^2$ .

Step 7: Round the calculated acceleration to match the given options:  $1.2753 m/s^2$  rounds to  $1.28 m/s^2$ .

Step 8: The calculated acceleration matches option (C)  $1.28 m/s^2$ .

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

Figure 18. Case of AMATH-SFT.

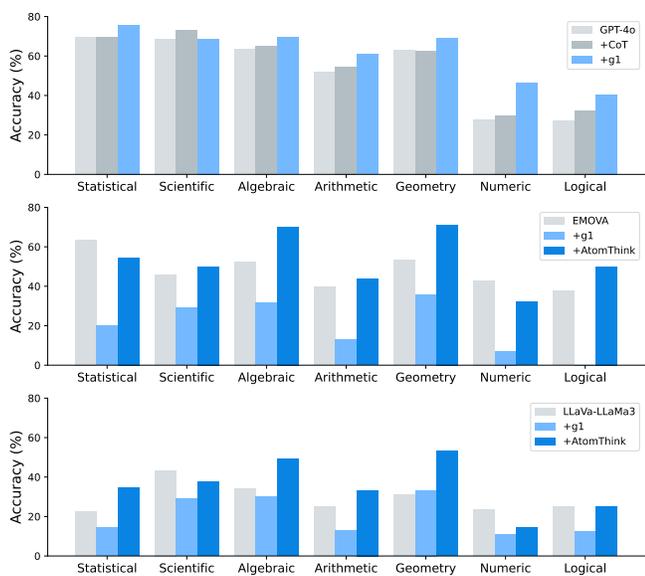


Figure 19. Comparison to CoT and g1 in MathVista subsets. In contrast to the declining trend observed in g1, AtomThink outperforms the baseline across most subsets.