TIR-Bench: A Comprehensive Benchmark for Agentic Thinking-with-Images Reasoning

Ming Li[1⋆], Jike Zhong[2⋆], Shitian Zhao[1⋆], Haoquan Zhang[1,4⋆], Shaoheng Lin[1⋆], Yuxiang Lai[3⋆], Chen Wei[5], Konstantinos Psounis[2], and Kaipeng Zhang[1⋆⋆]

[1] Shanghai AI Laboratory, Shanghai, China
[2] University of Southern California, Los Angeles, CA, USA
[3] Emory University, Atlanta, GA, USA
[4] The Chinese University of Hong Kong, Hong Kong SAR, China
[5] Rice University, Houston, TX, USA

Implementation Code Benchmark Data

Abstract. The frontier of visual reasoning is shifting toward models like OpenAI o3, which can intelligently create and operate tools to transform images for problem-solving, also known as thinking-*with*-images in chain-of-thought. Yet existing benchmarks fail to fully capture this advanced capability. Even Visual Search, the most common benchmark for current thinking-*with*-images methods, tests only basic operations such as localization and cropping, offering little insight into more complex, dynamic, and tool-dependent reasoning. We introduce **TIR-Bench**, a comprehensive benchmark for evaluating agentic thinking-with-images across 13 diverse tasks, each requiring novel tool use for image processing and manipulation in chain-of-thought. We evaluate 22 multimodal large language models (MLLMs), from leading open-sourced and proprietary models to those with explicit tool-use augmentation. Results show that TIR-Bench is universally challenging, and strong performance requires genuine thinking-with-images capabilities. Finally, we present a pilot study comparing direct versus agentic fine-tuning.

Keywords: Thinking-with-Image · Visual Reasoning · Benchmark

1 Introduction

The reasoning abilities of recent multimodal large language models (MLLMs) [4, 6] have advanced significantly, driven in large part by reasoning techniques such as chain-of-thought (CoT) [39]. By decomposing reasoning of complex visual questions into a series of textual steps, MLLMs are able to achieve improved performance. While promising, these techniques are confined to the textual domain, conducting their reasoning solely through language while treating the visual information as a static, unalterable input [28].

⋆ Equal contribution.
⋆⋆ Corresponding author.

To effectively process visual information, thinking-with-images has been proposed [5, 23, 28]. This approach enables a model to generate new visual information by actively manipulating input images with tools. For example, when faced with a complex visual problem, OpenAI's o3 model [23] first writes code to create an image-processing tool, then executes it to modify the image (*e.g.*, cropping, flipping, or rotating). The transformed visual data then informs the next stage of its linguistic reasoning.

To assess the agentic thinking-with-images capabilities of MLLMs, existing benchmarks [36, 40] have largely centered on visual search and tasks requiring the analysis of high-resolution images. These evaluations primarily validate a model's ability to accurately localize and crop specific regions within an image for better capturing detailed information to answer corresponding questions. However, these assessments tend to focus narrowly on the visual search capabilities of agentic MLLMs, leaving a broader spectrum of thinking-with-images abilities unevaluated. Therefore, there is an urgent need for a benchmark that incorporates a diverse range of tasks requiring sophisticated tool use to properly assess integrated multimodal reasoning.

In this paper, we introduce TIR-Bench, a comprehensive benchmark designed to evaluate the diverse thinking-with-images capabilities of agentic MLLMs. Unlike previous benchmarks focusing solely on visual search problems, TIR-Bench incorporates a diverse set of 13 tasks that require a wide range of tool-based interactions, such as zooming, rotating, increasing image contrast, adding auxiliary lines, and others to assess a model's tool integrated reasoning capabilities. The design of each task is predicated on the human intuition that solving it requires actively manipulating the visual input, rather than relying on static observation alone. For example, in TIR-Bench, a math problem might require the model to draw auxiliary lines or a coordinate system to find a solution, while a jigsaw puzzle task demands that it segment and then reassemble the image pieces. Consequently, TIR-Bench enables a more holistic evaluation of a model's thinking-with-images abilities, assessing a spectrum of skills not limited to visual search.

Using TIR-Bench, we conduct a comprehensive performance evaluation of 22 leading MLLMs across three categories: open-source models, proprietary models, and tool-using agents. The overall experimental results, illustrated in Figure 1, reveal that TIR-Bench is a challenging benchmark for thinking-with-images abilities, as the best performance achieved is only 46%. Moreover, traditional non-agentic models perform poorly on TIR-Bench, with the best-performing model Gemini-2.5-pro reaching an accuracy of merely 28.9%. These findings highlight the importance of the thinking-with-images ability for this benchmark, as models equipped with tool-use capabilities, such as o3, o4-mini, and PyVision [50], achieve much higher performance than other models.

Lastly, we assess the function-calling proficiency of various MLLMs and conduct a pilot study contrasting direct supervised fine-tuning (SFT) with an agentic SFT approach on rotated image OCR task. Many recent MLLMs are equipped with function-calling capabilities. Our evaluation on rotation game task of TIR-

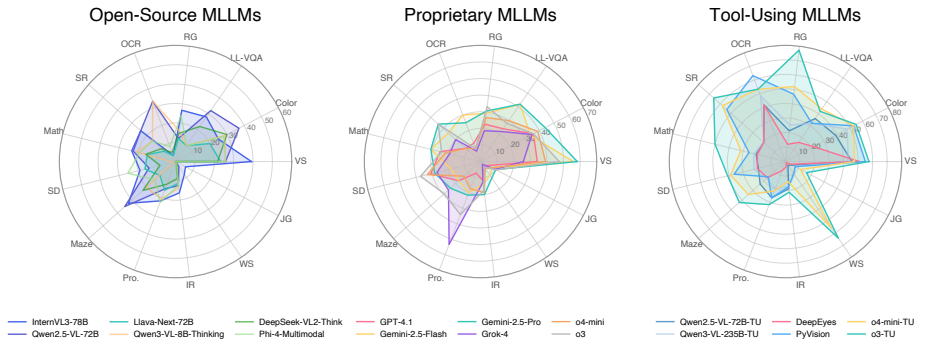


Fig. 1: Overview of performance of open models (left), proprietary models (middle), and agentic tool-using models (right). SR: Symbolic Reasoning, WS: Word Search, LL-VQA: Low-Light VQA, IR: Instrument Reasoning, SD: Spot Difference, JG: Jigsaw Game, VS: Visual Search, RG: Rotation Game, Pro.: Proportion VQA. o3-TU: o3-tool-using, i.e., o3 with code interpreter.

Bench measures a model's proficiency in accurately executing the correct tool parameters as part of its reasoning chain. Results show that recent models like o3 perform well, whereas earlier models such as GPT-4o perform significantly worse. The pilot study on the rotated image OCR task compares two training methodologies and examines whether end-to-end SFT can achieve strong performance across different data scales for tasks involving image operations. Our findings indicate that the agentic SFT on full problem-solving trajectories with generated images is significantly more effective than direct SFT on rotated OCR task. This implies that agentic fine-tuning enables the emergence of more complex and robust problem-solving behaviors, allowing models to tackle multi-step tasks that are intractable with direct fine-tuning alone.

2 Related Works

2.1 Multimodal Benchmarks

As the capabilities of Multimodal Large Language Models (MLLMs) evolve rapidly, a variety of benchmarks have been proposed to evaluate their performance, identify limitations, and guide future improvements [12, 13, 18, 19, 25, 26, 33, 38, 46]. These benchmarks are typically either specialized for specific domains [12, 13, 19, 38, 43, 52] or designed to be versatile and cover a broad range of tasks [11, 17, 37, 45]. MLLMs often use CoT reasoning on these benchmarks, though solving them only requires static image information. More recently, benchmarks such as V* Bench and HR-Bench have been introduced to evaluate agentic capabilities, specifically for visual search in high-resolution images [36, 40]. While these benchmarks advance agentic evaluation by requiring models to programmatically crop high-resolution images, their focus is narrowly confined to visual search. Consequently, a broader spectrum of thinking-with-images abilities, such as rotation and drawing, remains largely underexplored.

2.2 Think with Images

In previous works, Visual Sketchpad [5], CoGCoM [24], DeepEyes [51], Pixel Reasoner [27], OpenThinkIMG [28], Chain-of-Focus [47], Mini-o3 [9] and REVPT [53] generate and executes tool calling in a predefined visual-specific toolset. In the intermediate reasoning steps, processed images are reinjected to the context, resulting in a multi-modal rational. However, these methods rely on a fixed collection of external visual parsers—such as detection models (e.g., GroundingDINO [16,35]) and segmentation models (e.g., SAM [7,34])—which constrains their generality across diverse vision tasks and introduces bottlenecks due to dependency on external models. In contrast, ViperGPT [29], o3 [22], o4-mini, Thyme [48] and PyVision [44, 49, 50] adopt Python as its primitive tool. Capitalizing on the advanced coding and multimodal understanding capabilities of modern MLLMs—such as Claude-4.0 [2] and GPT-4.1 [21], enables the MLLM to dynamically write and execute code to construct complex, task-specific tools on demand, thereby supporting more general and flexible reasoning. This aligns with the emerging paradigm of "thinking with images" highlighted in o3's blog [23] as a powerful cognitive capability. To assess this important ability, we propose TIR-Bench, covering diverse tasks on which multi-modal rationals are necessary.

3 TIR-Bench

In this section, we introduce **TIR-Bench**. The overview of the benchmark is shown in Figure 2. We first introduce the task design strategy in subsection 3.1. Next, we introduce the data collection process in subsection 3.2. Finally, we present the benchmark summary in subsection 3.3.

3.1 Task Design

To extensively validate the model's ability to think with images we design 13 tasks. The benchmark's tasks are designed to evaluate a model's ability to perform active, tool-based visual reasoning, moving far beyond static image analysis. This includes tasks that require programmatic analysis, such as calculating color proportions or calling external models for object segmentation. Other challenges test the model's ability to overcome suboptimal conditions by programmatically enhancing low-light images or correcting the orientation of rotated text before performing OCR. The benchmark also features complex spatial and algorithmic puzzles, requiring models to solve mazes, reassemble jigsaw pieces, or draw auxiliary lines to solve geometric problems. Finally, it assesses fine-grained perception through tasks like spotting differences between images, reading instruments via cropping and zooming, and locating anomalies in visual puzzles. In every case, the model is forced to engage in a dynamic, multi-step process of visual manipulation and reasoning to arrive at the correct answer. More details about task design are introduced in Appendix.

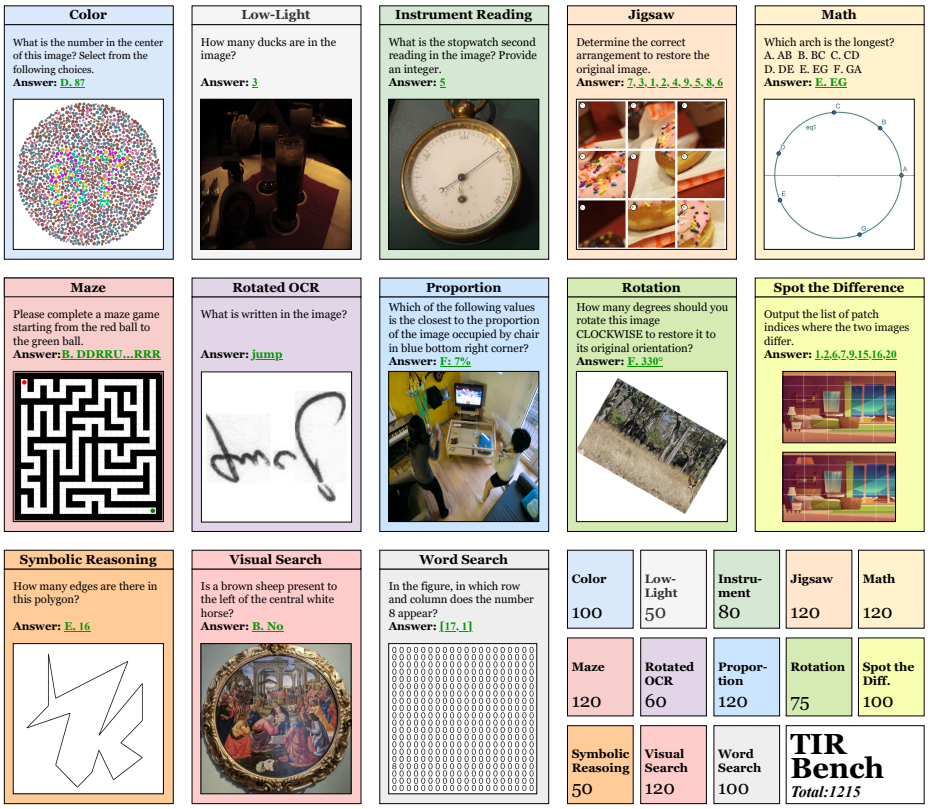


Fig. 2: Benchmark Overview. TIR-Bench is composed of 13 tasks, meticulously designed to evaluate a wide spectrum of thinking-with-images capabilities.

3.2 Data Collection

Guidelines. As previously mentioned, current benchmarks focus almost exclusively on visual search, overlooking a wide range of other tool-using abilities. To bridge this evaluation gap, we designed TIR-Bench in accordance with the following guidelines: (1) Diverse, Application-Grounded Tasks: TIR-Bench covers 13 distinct tasks across multiple domains, including spatial reasoning, visual perception, and mathematics, to mirror the complexity of real-world applications where static image analysis is insufficient. (2) Comprehensive Skill Assessment: The benchmark incorporates diverse visual contexts and requires a range of programmatic skills—from drawing auxiliary lines in geometry to executing pixel-level analysis—to foster a well-rounded evaluation of a model's agentic capabilities. (3) Probing Model Limitations: Each task is designed to be unsolvable without a multi-step, tool-based strategy. This intentional difficulty probes the limitations of current models, effectively distinguishing true thinking-with-images reasoning from simpler visual recognition. (4) Deterministic Evaluation: all tasks are designed with objectively verifiable answers, providing a robust framework for deterministic and reproducible evaluations. (5) Many samples in

TIR-Bench are newly annotated or generated, making it a more reliable benchmark that minimizes the risk of data contamination from models' pre-training corpora.

Collection. We briefly introduce the data collection process of the 13 tasks here, while the detailed process can be found in Appendix. The summary of data collection is shown in Table 1. Data from existing benchmarks are covered by the Apache License 2.0 or other licenses that permit academic use. For newly annotated data, images were sourced from freely available platforms like Wikimedia Commons and the National Gallery of Art, or were created by students using tools such as GeoGebra.

(1). Collection of Math VQA, Symbolic Reasoning, Low-Light VQA, and Instrument Reading tasks: Data from these tasks are newly created and annotated data. We tasked two Ph.D. students with sourcing free images from the Internet and MathVista [18] or create images using Geogebra, and then creating corresponding question-answer pairs. The annotators were explicitly instructed to design problems that, in their judgment, would necessitate tool-based image manipulation for a solution, ensuring the tasks could not be solved by static observation alone. Both students engaged in a reciprocal cross-checking process to verify the correctness of the problems and their respective answers.

(2). Collection of Color VQA task: We tasked a Ph.D. student with curating samples from the ColorBench [14], specifically selecting instances that could not be solved by static observation alone and therefore necessitate programmatic analysis to answer correctly. Finally, we select 100 problems from ColorBench [14].

(3). Collection of Jigsaw, Maze, Rotation, Word Search Tasks: The data for these tasks were programmatically generated to create controlled and scalable challenges. For the Jigsaw Puzzle, we selected 120 images from the RefCOCO dataset, chosen for their prominent objects. These images were then segmented into grids ranging from 3×3 to 6×6 and shuffled to create puzzles of varying difficulty. For the Maze task, we programmatically generated 100 mazes with sizes scaling from 5×5 to 62×62. For the Rotation Game, we utilized 75 images from CVBench [32], to which we applied random rotations. Three difficulty tiers were established based on the magnitude of the rotation angle (e.g., 5, 10, or 15 degrees). Finally, for the Word Search task, we programmatically generated 85 samples of different sizes and supplemented them with 15 complex puzzles sourced from the internet.

(4). Collection of Spot the Difference: We sourced pairs of nearly identical cartoon and real-world images from the internet, with one image in each pair containing subtle alterations. Both images were then segmented into an $m \times n$ grid of corresponding patches. Finally, two annotators reviewed the pairs to identify and label the specific patches that contained the differences. Both students engaged in a reciprocal cross-checking process to verify the correctness of difference patch numbers.

(5). Collection of Proportion VQA: For this task, we collected 120 images from the RefCOCO dataset and used the ground-truth segmentation masks

Attribute	Color VQA	Proportion VQA	Symbolic Reasoning	Math	Word Search	Rotated OCR
Image Source	ColorBench	RefCOCO	VlmsAreBlind + Web-sourced	Web-sourced	Web-sourced	OCRBench
QA Construction	Human-annotated	Synthetic	Human-annotated	Human-annotated	Synthetic	Dataset-provided
Samples	100	120	50	120	100	60
Answer Type	Single-choice	Single-choice	Single-choice	Single-choice	Open-ended	Open-ended

Attribute	Maze	Low-Light	Instrument Reading	Spot the Difference	Jigsaw	Visual Search	Rotation
Image Source	Synthetic	Web-sourced	Web-sourced	Web-sourced	Synthetic	Web-sourced	CVBench
QA Construction	Synthetic	Human-annotated	Human-annotated	Human-annotated	Synthetic	Synthetic	Synthetic
Samples	120	50	80	100	120	120	75
Answer Type	Single-choice	Open-ended	Open-ended	Open-ended	Open-ended	Single-choice	Single-choice

Table 1: Overview of all tasks, grouped into reasoning-oriented (top) and perception-oriented (bottom), with their sources, QA constructions, sample sizes, and answer types.

to calculate the correct object proportions. The incorrect multiple-choice options were then generated by adding or subtracting eight percentage points from the true value.

(6). Collection of Rotated Image OCR Task: We selected 60 images from OCRBench and applied a rotation to each. The rotation probabilities were 25% for 90°, 25% for 270°, and 50% for 180°.

(7). Collection of Visual Search: Recognizing that problems in existing benchmarks like V* Bench [40] are often too simple for current models, we curated a more challenging dataset for this task. We began by selecting 32 difficult problems from HR-Bench [36]. To supplement these, we collected 88 new samples, which include 25 high-resolution art images and 63 high-resolution real-world images sourced from the internet. For each of these new images, an annotator was tasked with generating a unique question-answer pair. In total, the Visual Search task comprises 120 samples, 88 of which are newly created for this benchmark.

3.3 Benchmark Summary

TIR-Bench consists of a total of 1215 examples divided into 13 diverse but essential tasks. Questions in our benchmark are categorized into two types: multi-choice and free-form problems, counting 665 and 550, respectively. The average length for question text is 46.29 and the average length for answer text is 3.41. The distribution of each task is shown in Appendix. Image and question examples for each task are shown in Appendix.

4 Experiments

In this section, we detail our evaluation setup and results for **22** leading MLLMs, considering both models with and without agentic tool-use capabilities. Our goal is to show that MLLMs lacking the ability to think with images perform

poorly on TIR-Bench. We describe the experimental setup in subsection 4.1, report the results in subsection 4.2, present experiments on function calling in subsection 4.5, and compare agentic SFT with end-to-end SFT in subsection 4.6.

4.1 Experiment Setup

Model selection. We categorize the MLLMs we use into open-source, proprietary, and tool-using. For open-sourced, we evaluated 11 models across three widely used and up-to-date model families: DeepSeek-VL-2 [41], Kwai Keye-VL [30], Phi-4-Multimodal [1], LLaVA [10, 15], Qwen2.5-VL [3], InternVL3 [54], and Qwen3-VL [31], ranging from 3B to 78B. Results from these models accurately reflect open-source MLLMs' performance on thinking-with-image reasoning tasks. For proprietary models, we selected 7 models across 3 model families: GPT [6], Gemini-2.5 [4], and Grok-4 [42]. We test GPT series, including GPT-4.1, GPT-4o [6], as well as o-series models [22], including o3 and o4-mini. For these GPT models, we use Azure API for calling models w/o python interpreter or sandbox [23]. For agentic tool-using (TU) MLLMs, we evaluate three open-sourced frameworks: DeepEyes [51], and PyVision [50], and 2 proprietary models o4-mini-TU and o3-TU. We use the official OpenAI API and turn on code interpreter and set the container parameter as auto.

Evaluation. We follow previous works [13,18] to first generate answers from models and subsequently using GPT-4o to extract the final answer from the answer content. For multiple-choice and short-form answers, we compare the extracted value directly against the ground-truth to calculate accuracy; for grounding type problems such as Jigsaw Game and Spot the Difference with list type answer, we calculate the intersection over union (IoU).

Implementation details. We conduct all evaluations in zero-shot manner for fair comparison and better generalization. For open models, all experiments are done on NVIDIA A100 GPUs. For proprietary models, we use the official API. More details can be found in the Appendix.

4.2 Experimental Results

Both average and task-wise accuracies are reported in Table 2. We discuss several findings below.

Result 1: TIR-Bench is challenging for all model types. The highest performance observed among all models is only 46%, a result that underscores the difficulty of the TIR-Bench. This benchmark proves to be a significant challenge for assessing Thinking-with-Images capabilities, even for advanced models such as o3-TU, which leverage a code interpreter.

Table 2: Model Accuracy (%) Across Various Evaluation Tasks. SR: Symbolic Reasoning, WS: Word Search, LL-VQA: Low-Light VQA, IR: Instrument Reasoning, SD: Spot Difference, JG: Jigsaw Game, VS: Visual Search, RG: Rotation Game, Pro.: Proportion VQA. o3-TU: o3-tool-using, i.e., o3 with code interpreter. o3: o3 without code interpreter.

Model	All	Color	Pro.	OCR	SR	Maze	Math	WS	LL-VQA	IR	SD	JG	VS	RG
Random Guess	13.5	28.0	6.7	-	14.0	13.3	15.8	0.0	4.0	8.8	22.6	5.8	22.5	16.0
Open-Source MLLMs														
DeepSeek-VL2-Direct	13.0	25.0	10.0	5.0	8.0	17.5	17.5	0.0	20.0	13.8	14.6	0.0	22.5	12.0
DeepSeek-VL2-Think	13.4	30.0	12.5	5.0	10.0	22.5	15.8	1.0	22.0	8.8	8.3	0.0	21.7	14.7
Keye-VL-8B	14.8	32.0	22.5	3.3	18.0	25.0	18.3	0.0	6.0	11.2	9.0	0.0	21.7	14.7
Phi-4-Multimodal	16.3	28.0	22.5	8.3	16.0	14.2	19.2	1.0	10.0	12.5	25.3	0.0	25.8	24.0
Qwen3-VL-8B-Instruct	16.9	30.0	10.0	41.7	14.0	30.8	18.3	2.0	20.0	16.2	9.8	0.0	22.5	14.7
Qwen3-VL-8B-Thinking	15.8	29.0	20.8	33.3	16.0	20.0	20.8	0.0	10.0	13.8	1.5	0.0	25.8	16.0
Llava-1.6-M-7B	11.3	27.0	7.5	3.3	16.0	4.2	16.7	0.0	14.0	6.3	18.0	0.0	22.5	12.0
Llava-1.6-V-7B	11.5	27.0	10.8	0.0	10.0	15.0	12.5	1.0	8.0	7.5	12.2	0.0	24.2	13.3
Llava-1.6-34B	13.0	31.0	6.7	1.7	20.0	15.8	18.3	0.0	16.0	16.3	11.9	0.0	21.7	10.7
Llava-Next-72B	11.2	20.0	15.8	3.3	8.0	10.8	15.0	0.0	10.0	11.3	16.3	0.0	23.3	12.0
Qwen2.5-VL-3B	17.7	27.0	20.0	31.7	12.0	21.7	20.8	0.0	12.0	13.8	29.7	0.0	26.7	12.0
Qwen2.5-VL-7B	16.0	21.0	10.8	48.3	14.0	15.0	24.2	0.0	22.0	11.3	24.4	0.0	21.7	9.3
Qwen2.5-VL-32B	18.7	26.0	19.2	25.0	10.0	18.3	23.3	2.0	14.0	15.0	13.1	5.4	48.3	13.3
Qwen2.5-VL-72B	19.7	37.0	15.0	33.3	24.0	35.0	22.5	3.0	32.0	12.5	14.1	0.0	25.8	12.0
InternVL3-8B	16.9	23.0	11.7	0.0	6.0	33.3	21.7	2.0	22.0	8.8	16.6	4.5	36.7	17.3
InternVL3-38B	19.1	24.0	10.8	3.3	26.0	23.3	29.2	8.0	28.0	13.8	14.6	5.1	44.2	13.3
InternVL3-78B	21.4	25.0	21.7	3.3	24.0	32.5	23.3	8.0	28.0	16.3	18.9	5.8	39.2	26.7
Proprietary MLLMs														
GPT-4.1	18.8	36.0	7.5	11.7	12.0	17.5	25.0	4.0	24.0	11.3	30.9	5.1	34.2	22.7
GPT-4o	17.3	26.0	22.5	10.0	10.0	20.0	15.8	0.0	26.0	7.5	19.4	6.2	35.0	20.0
Gemini-2.5-Flash	25.2	34.0	20.0	30.0	26.0	17.5	30.8	10.0	42.0	13.8	18.5	8.0	55.8	29.3
Gemini-2.5-Pro	28.9	44.0	21.7	25.0	34.0	24.2	30.8	12.0	42.0	20.0	28.5	10.4	**58.3**	30.7
Grok-4	22.5	35.0	**53.3**	6.7	20.0	25.8	19.2	2.00	22.0	12.5	27.0	10.0	25.8	18.7
o4-mini	21.2	39.0	17.5	8.3	12.0	13.3	21.7	5.0	30.0	18.8	33.0	8.0	39.2	26.7
o3	26.9	36.0	34.2	8.3	34.0	29.2	24.2	4.0	28.0	17.5	37.2	10.8	47.5	33.3
Tool-Using MLLMs														
Qwen2.5-VL-72B-TU	24.0	39.0	26.7	41.7	20.0	23.3	20.8	3.0	36.0	18.8	19.2	5.3	45.8	21.3
Qwen3-VL-235B-TU	24.9	48.0	20.0	48.3	20.0	17.5	22.5	11.0	32.0	17.5	22.4	7.3	43.3	25.3
DeepEyes	17.3	22.0	6.7	41.7	19.9	16.7	20.0	1.0	16.0	3.8	19.9	3.9	50.8	12.0
PyVision	31.8	53.0	26.7	**63.3**	54.0	15.8	25.8	10.0	32.0	17.5	36.4	7.6	55.0	46.7
o4-mini-TU	37.5	53.0	21.7	53.3	58.0	34.2	31.7	55.0	**44.0**	13.8	38.9	11.8	47.5	52.0
o3-TU	**46.0**	**55.0**	31.7	53.3	**66.0**	**42.5**	**50.0**	**64.0**	42.0	**21.3**	**41.0**	**16.4**	57.5	**77.3**

Result 2: Traditional non-agentic models perform poorly on TIR-Bench. Across all tasks, non-tool-using MLLMs show poor performance: the performance of most open-source models is close to or slightly higher than the random guess performance while the top-performing MLLM, Gemini-2.5-pro, surpasses random guess results by only 15%. These results highlight that, without agentic tool-using abilities, MLLMs can not perform well on TIR-Bench.

Result 3: Agentic tool-using is essential for TIR-Bench. The o3-TU model [22] demonstrates the strongest overall performance, achieving the highest average accuracy at 46%. This represents a substantial lead, outperforming the Gemini-2.5-Pro model by nearly 17% and the o3 model without a code interpreter by

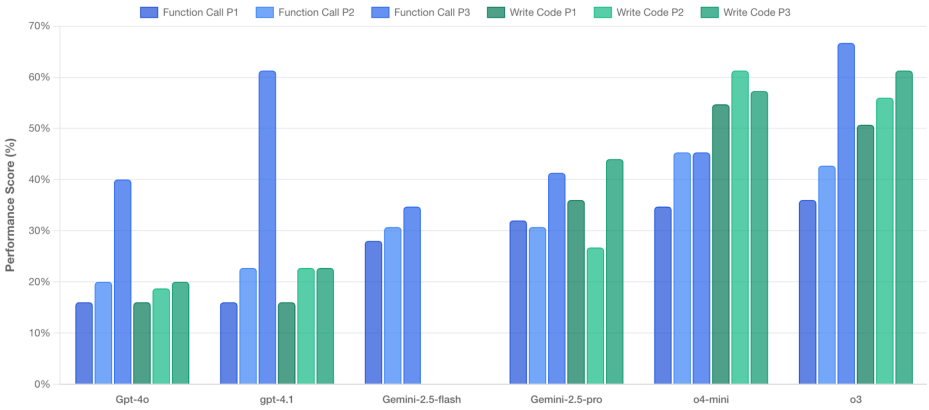


Fig. 3: Comparison of function-calling and code-writing abilities across models. Function calling uses a predefined rotation function, while code writing requires the model to implement it. P1 includes only the question; P2 adds a hint to use the function; P3 explicitly instructs testing each degree in the answer choices. Stronger models (e.g., o3) perform better at code writing, whereas weaker ones (e.g., GPT-4o) perform better at function calling.

19%. With tool-use enabled, it achieves state-of-the-art results on the majority of the tasks, winning 10 out of the total 14 categories. The o3-TU and o4-TU models demonstrate a large performance improvement on tasks involving straightforward image manipulation or processing, including the rotation game, rotated image OCR, and word search. Similar phenomenon appears in PyVision, which implements thinking-with-images abilities based on GPT-4.1. PyVision brings 13% accuracy improvement compared with GPT-4.1.

We also observe that, although o3-TU excels in most categories, the improvements are not uniform. For example, for complex tasks such as *Jigsaw game*, the performance of o3-TU is still very low. On *Proportion VQA*, performance surprisingly decreases from 34.2% to 31.7%. This task requires calling external segmentation models such as Segment Anything [8] to obtain a good rough estimate of the object segments. However, the current o3-TU model can only write executable code to manipulate images, but lacks the capability to call segmentation models specifically.

4.3 Qualitative Analysis

We present several examples of o3-TU responses here and examples of other model responses in Appendix. Since the responses from the OpenAI API contain no reasoning process, we re-ran the questions in the web-based ChatGPT interface and analyzed the responses. Figure 4 shows an case of instrument reading. The model repeatedly crops the region around the pointer, checks its position, and guesses the answer (highlighted in blue). After several cycles of cropping,

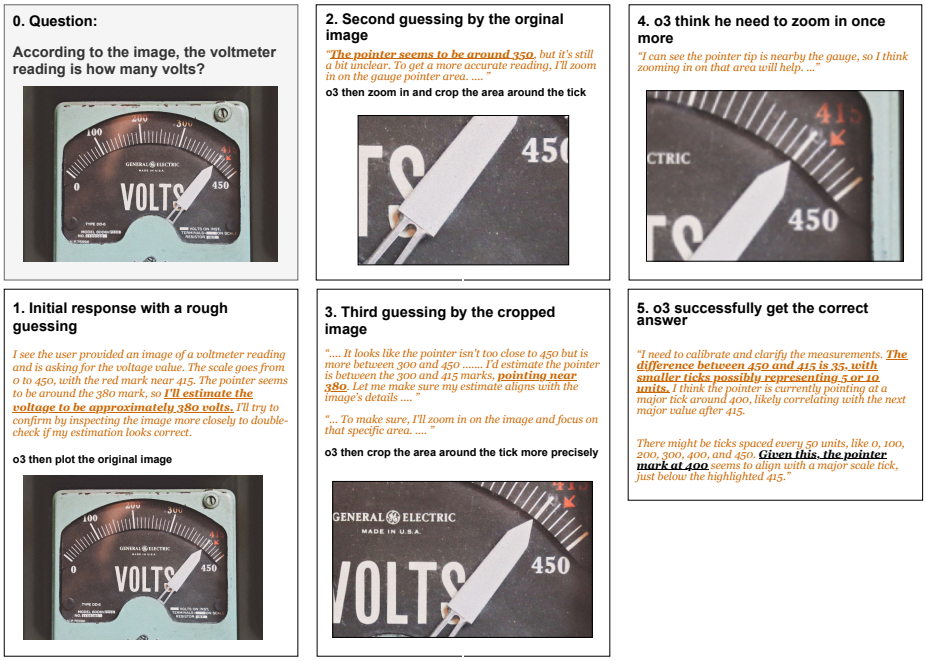


Fig. 4: An example of o3 solving an case of instrument reading from TIR-Bench.

observing, and reflecting, o3 finally confirms its answer as "400," which is correct. Figure 5 shows a case of jigsaw puzzle involved a lot of ineffective attempts. The model first divided the complete image into segments, then tried to compare and judge similarity by "understanding" the image as a whole. In the early empirical stage, the model made nearly 55 iterations, but failed to obtain any usable partial results. After that, it began to use edge comparison, sampling the borders of image pieces and comparing them pair by pair, which also did not produce effective outcomes. The model then started to develop algorithms, applying brute-force permutations to explore possible arrangements and calculate similarity. However, the initial algorithm did not work. After modifying the sampling method, the model, through more than 36,000 attempts, achieved a high similarity score that corresponded to the correct solution. This demonstrates that relying solely on the model's raw visual capabilities is insufficient; only code-based perception proves to be reliable.

4.4 Error and Efficiency Analysis

Due to the lack of access to the internal reasoning of OpenAI's o3 and o4-mini (as paper examples are sourced from the web interface), we utilize PyVision for this study. Analysis shows that 37% of PyVision's errors stem from perception issues, such as misinterpreting visual content or misreading instrument values via incorrect cropping. Another 61% arise from tool-generation and reasoning failures, including flawed logic in image-processing code; for instance, in jigsaw

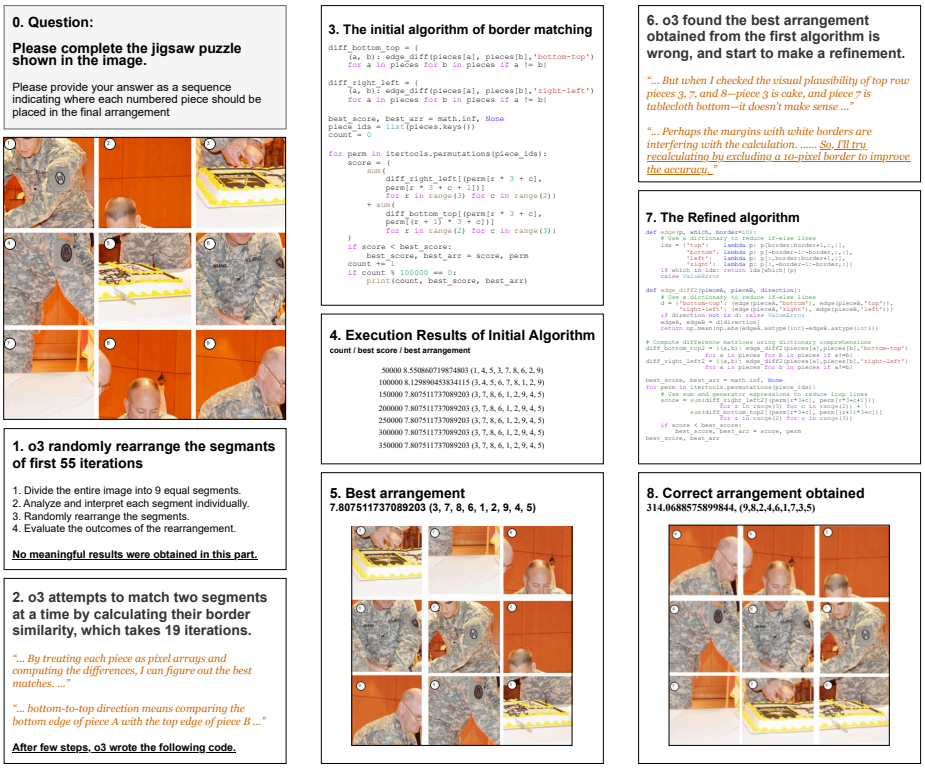


Fig. 5: The o3 model's reasoning process for a jigsaw puzzle task in TIR-Bench involved a large number of ineffective attempts. To describe o3's reasoning process more clearly and concisely, we summarize its key behaviors as **TITLES** for each stage in the illustration. Important thought processes are highlighted in orange, and the `main code` implementations along with their corresponding output results are provided.

tasks, the model may incorrectly split or rearrange patches, failing to solve the problem. The remaining 2% are miscellaneous.

We evaluate the agentic framework's efficiency by analyzing interaction turns for PyVision (Figure 6), which averages 4.72 turns per problem. We observe an inverse correlation between interaction length and success: mastered tasks like Rotated OCR and Color VQA exhibit efficient reasoning, requiring only 3.12 and 3.23 turns to achieve 63.3% and 53.0% accuracy, respectively. Conversely, high interaction overhead typically signals error loops rather than productive reasoning; the Maze (8.58 turns, 15.8% performance) and Word Search (7.34 turns, 10.0% performance) tasks incur the highest costs yet yield the lowest results. This suggests that while the agentic loop facilitates self-correction, interactions exceeding ∼5 turns often reflect the model struggling to find a valid solution path.

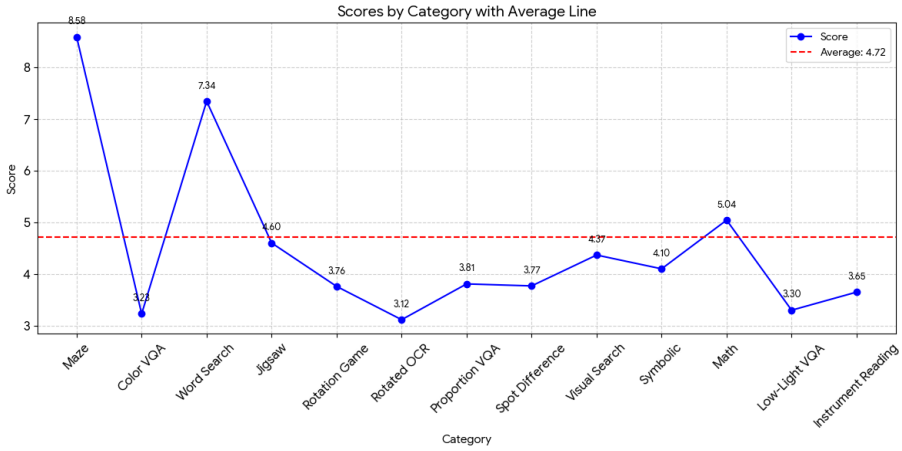


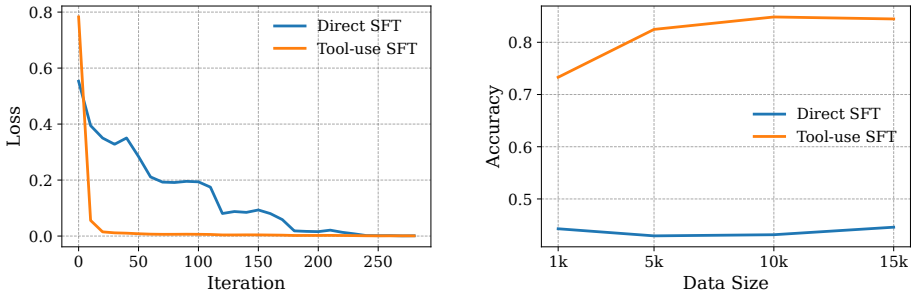
Fig. 6: Number of interaction turns for the PyVision model.

4.5 Function Call Experiment Results

In this subsection, we report the results on the function calling ability of different MLLMs.

Set-up. Using the Rotation Game task as a case study, we experimented with two distinct function-calling strategies: (1) providing the model with a predefined rotate function, requiring it only to output the degree parameter; and (2) requiring the model to generate the full image rotation code itself. Additionally, we tested three prompt variations to assess the impact of guidance: (1) a baseline prompt containing only the question (Prompt 1); (2) a prompt including a hint to leverage the rotation function (Prompt 2); and (3) a more explicit prompt instructing the model to systematically test each degree from the answer choices (Prompt 3). The specific function definitions and prompts used in this analysis are provided in Appendix. During inference, we repeatedly call functions until the model produces a final answer without requiring any function parameters.

Results. We report the experimental results in Figure. 3 and report the average number of calling for each problem in Appendix. Since Gemini-2.5-flash does not write code for all three prompts, we do not report it for writing code. We brief discuss some key findings here: **(1).** Clear Performance Hierarchy. o3 emerges as the top performer, achieving the highest accuracy, with o4-mini following closely. **(2).** Prompting Strategy is Key. The prompt strategy hinting to check each degree choice (prompt 3) works best. The performance of most models increases with prompt 3 compared to the other two prompts. This implies that models may not know how to best utilize the functions without explicit guidance. **(3).** Increased function call number in Newer Models. The average number of function calls for more recent advanced models (e.g., o3) is much higher than for previous models (e.g., GPT-4o). This implies that recent models are better trained for iterative function calling.



(a) Loss **(b)** Accuracy

Fig. 7: Comparison of change of loss and accuracy between direct SFT and tool-use SFT.

4.6 Fine-Tuning Comparison Experiment Results

In this subsection, we report the experimental results on the different fine-tuning strategies on rotated OCR task.

Setup. We use the rotated image OCR task as a case study to evaluate data scaling performance. We created training sets of varying sizes: 1k, 5k, 10k, and 15k samples—by randomly selecting and rotating images from the OCR-Dataset [20]. We then compared two distinct training strategies: (1). Direct SFT: A standard supervised fine-tuning approach where the model is trained to map the rotated image directly to the ground-truth text. (2). Tool-Use SFT: An agentic approach where the model first learns to output the correct rotation degree. The restored image is then concatenated with the original context, and the model is subsequently trained to read the text from this corrected visual input. We use Qwen-2.5-VL-7B and fully fine-tune all parameters with 5 epochs.

Results. We report accuracy on the Rotated OCR task in Figure 7b and the loss curves on 15k samples in Figure 7a. Overall, Tool-use SFT significantly outperforms Direct SFT. For Tool-use SFT, performance scales positively with data size, whereas Direct SFT shows no such trend. This suggests that simply scaling data for Direct SFT is ineffective on tasks requiring image-based reasoning. We also observe that Tool-use SFT's loss decreases much faster despite starting from a higher initial value. This is likely because Qwen-2.5-VL was not pretrained with function-calling, leading to higher initial loss. Furthermore, since Qwen-2.5-VL was trained mostly on correctly oriented OCR data, fine-tuning it directly on rotated data may cause forgetting. In contrast, restoring image orientation before text output avoids this issue, which may explain the faster loss reduction.

5 Conclusion

In this paper, we propose TIR-Bench, a comprehensive benchmark designed to evaluate the thinking-with-images ability of agentic MLLMs. TIR-Bench is composed of 13 meticulously collected tasks that assess a wide range of tool-assisted

reasoning skills. By assessing diverse MLLMs on TIR-Bench, including both standard models and those augmented with tool- use capabilities, we find that TIR-Bench is a challenging benchmark for all models that necessitates thinking-with-images capabilities for successful completion. Lastly, we conduct a pilot study comparing direct and agentic fine-tuning for image-operation tasks and evaluating the function-calling abilities of various MLLMs.

6 Acknowledgment

This research is supported in part by grants from the National Science Foundation (NSF) under grant number 1956435.

References

1. Abouelenin, A., Ashfaq, A., Atkinson, A., Awadalla, H., Bach, N., Bao, J., Benhaim, A., Cai, M., Chaudhary, V., Chen, C., et al.: Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. arXiv preprint arXiv:2503.01743 (2025)
2. Anthropic: Introducing claude 4 (2025), https://www.anthropic.com/news/claude-4
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Gemini Team, G.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint (arXiv:2507.06261) (2025)
5. Hu, Y., Shi, W., Fu, X., Roth, D., Ostendorf, M., Zettlemoyer, L., Smith, N.A., Krishna, R.: Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In: NeurIPS (2024)
6. Hurst, A., OpenAI: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024). https://doi.org/10.48550/arXiv.2410.21276, https://arxiv.org/abs/2410.21276, "o" for omni — large multimodal model accepting text, audio, image, and video inputs
7. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023), https://arxiv.org/abs/2304.02643
8. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023), https://arxiv.org/abs/2304.02643
9. Lai, X., Li, J., Li, W., Liu, T., Li, T., Zhao, H.: Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. arXiv preprint arXiv:2509.07969 (2025)
10. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
11. Li, B., Ge, Y., Ge, Y., Wang, G., Wang, R., Zhang, R., Shan, Y.: Seed-bench: Benchmarking multimodal large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13299–13308 (2024)
12. Li, K., Tian, Y., Hu, Q., Luo, Z., Ma, J.: Mmcode: Evaluating multi-modal code large language models with visually rich programming problems. arXiv preprint arXiv:2404.09486 (2024)
13. Li, M., Zhong, J., Chen, T., Lai, Y., Psounis, K.: Eee-bench: A comprehensive multimodal electrical and electronics engineering benchmark. arXiv preprint arXiv:2411.01492 (2024)

14. Liang, Y., Li, M., Fan, C., Li, Z., Nguyen, D., Cobbina, K., Bhardwaj, S., Chen, J., Liu, F., Zhou, T.: Colorbench: Can vlms see and understand the colorful world? a comprehensive benchmark for color perception, reasoning, and robustness. arXiv preprint arXiv:2504.10514 (2025)
15. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26296–26306 (2024)
16. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: ECCV (2024)
17. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)
18. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
19. Lu, P., Gong, R., Jiang, S., Qiu, L., Huang, S., Liang, X., Zhu, S.C.: Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. arXiv preprint arXiv:2105.04165 (2021)
20. Minh: Combined ocr dataset. Hugging Face Datasets (2024), https://huggingface.co/datasets/ducto489/ocr_datasets
21. OpenAI: Introducing gpt-4.1 in the api (2025), https://openai.com/index/gpt-4-1/
22. OpenAI: Openai o3: A reasoning model trained to think before responding. Blog post / system card (April 2025), introducing OpenAI o3 and o4-mini
23. OpenAI: Thinking with images (2025), https://openai.com/index/thinking-with-images/
24. Qi, J., Ding, M., Wang, W., Bai, Y., Lv, Q., Hong, W., Xu, B., Hou, L., Li, J., Dong, Y., Tang, J.: Cogcom: A visual language model with chain-of-manipulations reasoning. In: ICLR (2025)
25. Qiao, R., Tan, Q., Dong, G., Wu, M., Sun, C., Song, X., GongQue, Z., Lei, S., Wei, Z., Zhang, M., et al.: We-math: Does your large multimodal model achieve human-like mathematical reasoning? arXiv preprint arXiv:2407.01284 (2024)
26. Qiu, Y., Zhang, H., Xu, Z., Li, M., Song, D., Wang, Z., Zhang, K.: Ai idea bench 2025: Ai research idea generation benchmark. arXiv preprint arXiv:2504.14191 (2025)
27. Su, A., Wang, H., Ren, W., Lin, F., Chen, W.: Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning (2025), https://arxiv.org/abs/2505.15966
28. Su, Z., Xia, P., Guo, H., Liu, Z., Ma, Y., Qu, X., Liu, J., Li, Y., Zeng, K., Yang, Z., et al.: Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. arXiv preprint arXiv:2506.23918 (2025)
29. Surís, D., Menon, S., Vondrick, C.: Vipergpt: Visual inference via python execution for reasoning. In: ICCV (2023)
30. Team, K.K., Yang, B., Wen, B., Liu, C., Chu, C., Song, C., Rao, C., Yi, C., Li, D., Zang, D., et al.: Kwai keye-vl technical report. arXiv preprint arXiv:2507.01949 (2025)
31. Team, Q.: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388
32. Tong, P., Brown, E., Wu, P., Woo, S., IYER, A.J.V., Akula, S.C., Yang, S., Yang, J., Middepogu, M., Wang, Z., et al.: Cambrian-1: A fully open, vision-centric ex-

ploration of multimodal llms. Advances in Neural Information Processing Systems **37**, 87310–87356 (2024)
33. Wang, K., Pan, J., Shi, W., Lu, Z., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset (2024), https://arxiv.org/abs/2402.14804
34. Wang, R.F., Cui, K., Schardong, I.B., Bauer, M.C., Somala, R.V., Xu, M., West, D., Jones, D.C., Taylor, S.V., Roberts, P.M., Li, C., Chee, P.W.: Jassidnet as a quantization-aware lightweight phenotyping framework for high-throughput cotton jassid (Amrasca biguttula) detection and counting toward objective resistance screening. Industrial Crops and Products **249**, 123716 (2026)
35. Wang, R.F., Petti, D., Chen, Y., Li, C.: Dinov3 visual representations for blueberry perception toward robotic harvesting. arXiv preprint arXiv:2603.02419 (2026)
36. Wang, W., Ding, L., Zeng, M., Zhou, X., Shen, L., Luo, Y., Yu, W., Tao, D.: Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 7907–7915 (2025)
37. Wang, X., Hu, Z., Lu, P., Zhu, Y., Zhang, J., Subramaniam, S., Loomba, A.R., Zhang, S., Sun, Y., Wang, W.: Scibench: Evaluating college-level scientific problem-solving abilities of large language models. arXiv preprint arXiv:2307.10635 (2023)
38. Wang, Z., Xia, M., He, L., Chen, H., Liu, Y., Zhu, R., Liang, K., Wu, X., Liu, H., Malladi, S., et al.: Charxiv: Charting gaps in realistic chart understanding in multimodal llms. arXiv preprint arXiv:2406.18521 (2024)
39. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)
40. Wu, P., Xie, S.: V?: Guided visual search as a core mechanism in multimodal llms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13084–13094 (2024)
41. Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., Dai, D., Gao, H., Ma, Y., Wu, C., Wang, B., et al.: Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. arXiv preprint arXiv:2412.10302 (2024)
42. xAI: Grok 4: Native tool use, real-time search integration, and frontier reasoning. xAI blog post / documentation (July 2025), includes the "Grok 4 Heavy" variant, 256 000 token context window, and performance improvements via reinforcement learning and expanded training data
43. Xiong, T., Ge, Y., Li, M., Zhang, Z., Kulkarni, P., Wang, K., He, Q., Zhu, Z., Liu, C., Chen, R., et al.: Multi-crit: Benchmarking multimodal judges on pluralistic criteria-following. arXiv preprint arXiv:2511.21662 (2025)
44. Yao, M., Huo, Y., Ran, Y., Tian, Q., Wang, R., Wang, H.: Neural Radiance Field-Based Visual Rendering: A Comprehensive Review . IEEE Transactions on Visualization & Computer Graphics **32**(07), 7361–7379 (2026). https://doi.org/10.1109/TVCG.2026.3677182
45. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9556–9567 (2024)
46. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Gao, P., et al.: Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? arXiv preprint arXiv:2403.14624 (2024)

47. Zhang, X., Gao, Z., Zhang, B., Li, P., Zhang, X., Liu, Y., Yuan, T., Wu, Y., Jia, Y., Zhu, S.C., Li, Q.: Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl (2025), https://arxiv.org/abs/2505.15436
48. Zhang, Y.F., Lu, X., Yin, S., Fu, C., Chen, W., Hu, X., Wen, B., Jiang, K., Liu, C., Zhang, T., et al.: Thyme: Think beyond images. arXiv preprint arXiv:2508.11630 (2025)
49. Zhao, S., Lin, S., Li, M., Zhang, H., Peng, W., Zhang, K., Wei, C.: Pyvision-rl: Forging open agentic vision models via rl. arXiv preprint arXiv:2602.20739 (2026)
50. Zhao, S., Zhang, H., Lin, S., Li, M., Wu, Q., Zhang, K., Wei, C.: Pyvision: Agentic vision with dynamic tooling. arXiv preprint arXiv:2507.07998 (2025)
51. Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C., Yu, X.: Deepeyes: Incentivizing "thinking with images" via reinforcement learning (2025), https://arxiv.org/abs/2505.14362
52. Zhou, P., Peng, X., Zhang, F., Xu, Z., Ai, J., Qiu, Y., Zhao, W., Song, J., Li, C., Tang, W., et al.: Mdk12-bench: a multi-discipline benchmark for evaluating reasoning in multimodal large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 28982–28990 (2026)
53. Zhou, Z., Chen, D., Ma, Z., Hu, Z., Fu, M., Wang, S., Wan, Y., Zhao, Z., Krishna, R.: Reinforced visual perception with tools. arXiv preprint arXiv:2509.01656 (2025)
54. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)