

# MathCanvas: Intrinsic Visual Chain-of-Thought for Multimodal Mathematical Reasoning

Weikang Shi<sup>1\*</sup> Aldrich Yu<sup>1\*</sup> Rongyao Fang<sup>1\*†</sup> Houxing Ren<sup>1</sup> Ke Wang<sup>1</sup> Aojun Zhou<sup>1</sup> Changyao Tian<sup>1</sup>  
Xinyu Fu<sup>2</sup> Yuxuan Hu<sup>1</sup> Zimu Lu<sup>1</sup> Linjiang Huang<sup>5</sup> Si Liu<sup>5</sup> Rui Liu<sup>2‡</sup> Hongsheng Li<sup>1,3,4‡</sup>

<sup>1</sup>CUHK MMLab, <sup>2</sup>Huawei Research, <sup>3</sup>Shenzhen Loop Area Institute, <sup>4</sup>CPII under InnoHK, <sup>5</sup>BUAA

wkshi@link.cuhk.edu.hk hsl@ee.cuhk.edu.hk

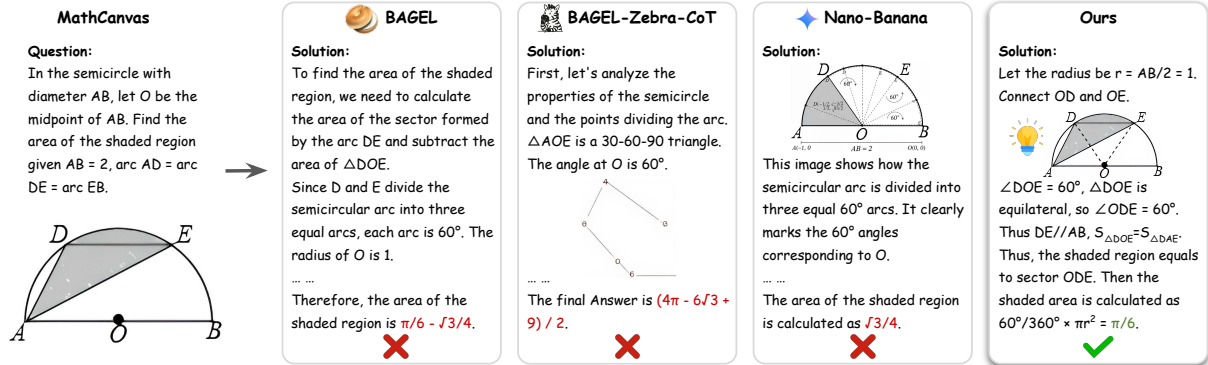


Figure 1: MathCanvas demonstrates the first successful application of intrinsic Visual Chain-of-Thought (VCoT) for complex mathematical reasoning. Prior attempts fail by generating incorrect (BAGEL-Zebra-CoT) or strategically poor (Nano-Banana) visuals, leading to wrong solutions. In contrast, MathCanvas correctly generates an intermediate visual step that unlocks a simpler, elegant solution path.

## Abstract

While Large Language Models (LLMs) have excelled in textual reasoning, they struggle with mathematical domains like geometry that intrinsically rely on visual aids. Existing approaches to Visual Chain-of-Thought (VCoT) are often limited by rigid external tools or fail to generate the high-fidelity, strategically-timed diagrams necessary for complex problem-solving. To bridge this gap, we introduce MathCanvas, a comprehensive framework designed to endow unified Large Multimodal Models (LMMs) with intrinsic VCoT capabilities for mathematics. Our approach consists of two phases. First, a *Visual Manipulation* stage pre-trains the model on a novel 15.2M-pair corpus, comprising 10M caption-to-diagram pairs (MathCanvas-Imagen) and 5.2M step-by-step editing trajectories (MathCanvas-Edit), to master diagram generation and editing. Second, a *Strategic Visual-Aided Reasoning* stage fine-tunes the model on MathCanvas-Instruct, a new 219K-example dataset of interleaved visual-textual reasoning paths, teaching it *when* and

*how* to leverage visual aids. To facilitate rigorous evaluation, we introduce MathCanvas-Bench, a challenging benchmark with 3K problems that require models to produce interleaved visual-textual solutions. Our model, BAGEL-Canvas, trained under this framework, achieves an 86% relative improvement over strong LMM baselines on MathCanvas-Bench, demonstrating excellent generalization to other public math benchmarks. Our work provides a complete toolkit—framework, datasets, and benchmark—to unlock complex, human-like visual-aided reasoning in LLMs. Project Page: <https://mathcanvas.github.io/>

## 1 Introduction

Mathematical reasoning represents a pinnacle of human intelligence, demanding a sophisticated interplay of logical deduction, symbolic manipulation, and abstract thinking. The advent of Large Language Models (LLMs) (DeepSeek-AI et al., 2025; Yang et al., 2024; OpenAI et al., 2024b) has marked a significant milestone in artificial intelligence, demonstrating remarkable capabilities in tackling complex mathematical reasoning tasks. A key driver of recent progress in LLM-based reasoning has been the Chain-of-Thought (CoT) (Wei

\*Equal Contribution

†Project lead

‡Corresponding author

et al., 2023) technique, which enables models to externalize intermediate steps and significantly improves performance on mathematical tasks.

However, the purely textual nature of CoT presents a fundamental limitation in domains like geometry and function analysis, where human problem-solving intrinsically involves constructing and manipulating visual aids, and even state-of-the-art models struggle in its absence (see Figure 12 in Appendix D). This gap has motivated the development of Visual Chain-of-Thought (VCoT), which aims to integrate visual information into the reasoning process. Early approaches to VCoT have predominantly relied on external specialized tools, such as dedicated vision models (Shao et al., 2024a; Hu et al., 2024; Gao et al., 2025b) or code interpreters (Hu et al., 2024; Wang et al., 2025c,d). While effective in specific contexts, these tool-based methods are often rigid, constrained to a predefined set of operations, and dependent on specific input formats (e.g., source code), which hinders their flexibility and broader applicability. Recent work has explored intrinsic VCoT, where unified large multimodal models (LMMs) natively generate visual thoughts as an integral part of their reasoning process (Cheng et al., 2025; Li et al., 2025b,a; Chern et al., 2025).

Though promising, these previous attempts have been confined to simple domains and have yet to succeed in mathematics due to two key challenges. First, current unified LMMs lack the capability to generate and iteratively edit the high-fidelity mathematical diagrams required for precise reasoning. The generated visuals are often geometrically incorrect, rendering them useless for logical deduction, as shown with BAGEL-Zebra-CoT (Li et al., 2025a) in Figure 1. Second, and more fundamentally, models lack the procedural knowledge to employ visual aids as a strategic component of their reasoning process—the complex decision of determining *when* to draw, *what* to draw, and *how* to leverage the visualization for subsequent logical deduction. This strategic failure is evident even in advanced models like Nano-Banana (Comanici et al., 2025), shown in Figure 1, whose generated visual acts more as a flawed decoration than an integral reasoning step, ultimately failing to uncover the key insight needed for the solution.

To this end, we argue that addressing these challenges requires models capable of interleaving textual deduction with the creation and modification of visual aids. Accordingly, we introduce **Math-**

**Canvas**, a comprehensive framework designed to endow unified LMMs with intrinsic VCoT capabilities for complex mathematical problem-solving. Our approach is structured around two complementary phases: *Visual Manipulation* and *Strategic Visual-Aided Reasoning*.

The first phase, *Visual Manipulation*, focuses on equipping the model with foundational visual synthesis and editing skills. To achieve this, we construct a new million-scale pretraining corpus specifically for mathematical diagrams. This resource comprises two parts: **MathCanvas-Edit**, containing 5.2M step-by-step diagram editing instruction pairs generated via a hybrid pipeline that combines LLM-driven mining with programmatic synthesis, and **MathCanvas-Imagen**, with 10M caption-to-diagram pairs. Pretraining on them imparts the robust diagram generation and manipulation abilities that form the bedrock of our approach.

The second phase, *Strategic Visual-Aided Reasoning*, aims to teach the model how to interleave diagrammatic actions with its textual reasoning steps. For this purpose, we curate **MathCanvas-Instruct**, the first large-scale dataset for interleaved visual–textual mathematical reasoning. It contains 219K training examples, where each solution is represented as an interleaved sequence of textual reasoning and corresponding visual steps. As demonstrated in Figure 1, training on **MathCanvas-Instruct** enables the model to learn how to coordinate diagrammatic actions with reasoning trajectories to successfully solve complex problems.

Furthermore, to rigorously evaluate models’ capabilities in visual–textual mathematical reasoning, we introduce a dedicated benchmark test set **MathCanvas-Bench** comprising 3K carefully curated problems. Each test instance requires the solver to produce coherent interleaved reasoning and visual outputs. We benchmarked 20 leading LMMs on this dataset, revealing substantial performance gaps and establishing it as a challenging and comprehensive testbed for future research on Visual Chain-of-Thought reasoning.

In summary, our contributions are as follows:

- We propose **MathCanvas**, a comprehensive framework that enables LMMs to perform **intrinsic VCoT** reasoning for complex mathematical problem solving.
- We construct two large-scale corpora tailored for our two-phase approach: a 15.2M-pair pretraining dataset for *Visual Manipulation*, and a

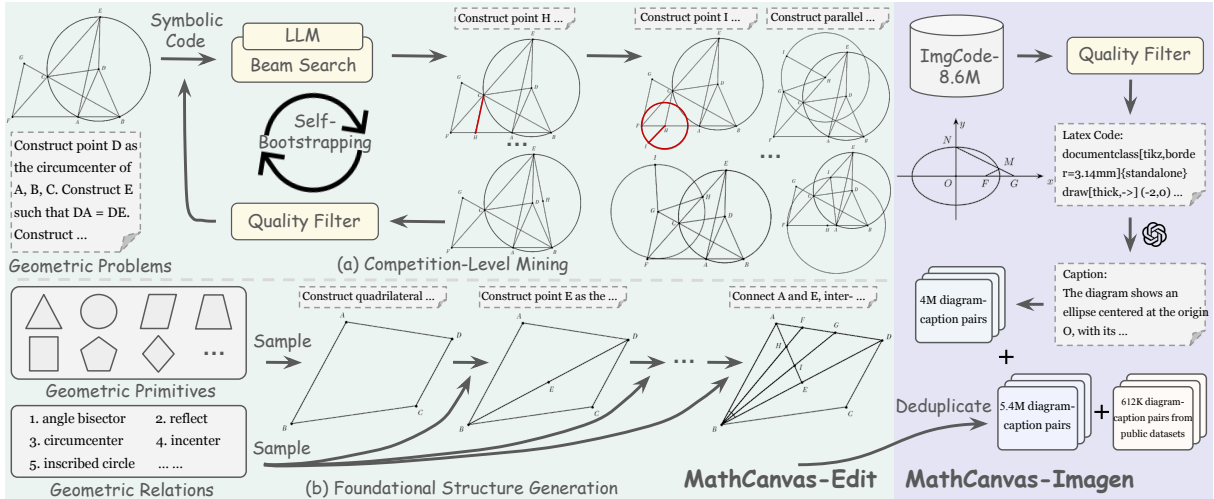


Figure 2: Curation pipeline for MathCanvas-Edit (left) and MathCanvas-Imagen (right). **(a) Competition-Level Mining:** An LLM with beam search proposes auxiliary constructions for competition problems; surviving proposals yield 2–3 step trajectories where each diagram adds one new element (red). **(b) Foundational Structure Generation:** Starting from a sampled primitive (e.g., a quadrilateral), we iteratively append geometric relations to produce up to four progressively enriched diagrams covering fundamental editing operations. **MathCanvas-Imagen** aggregates three sources: deduplicated intermediate frames from MathCanvas-Edit, filtered and re-captioned ImgCode-8.6M, and 612K pairs from public datasets.

219K-example fine-tuning dataset for *Strategic Visual-Aided Reasoning*.

- We further introduce a dedicated MathCanvas-Bench test set with 3K problems and benchmark 20 leading LMMs on it, revealing substantial deficiencies and establishing a challenging evaluation bed for future research.
- Experiments show that our model trained under the MathCanvas framework achieves a **86% relative improvement** over strong LMM baselines on MathCanvas-Bench, demonstrating the effectiveness of our approach in unlocking intrinsic VCoT capabilities.

## 2 Related Work

**Mathematical Reasoning with Large Multimodal Models.** The remarkable success of text-only LLMs in mathematical reasoning, often driven by sophisticated chain-of-thought prompting (Wei et al., 2023; Yang et al., 2024; Yue et al., 2023; Wang et al., 2023; Shao et al., 2024b), has naturally spurred interest in extending these capabilities to the multimodal domain. Initial efforts in this area have largely involved adapting LMMs by enhancing vision-text alignment on domain-specific data and then fine-tuning on mathematical question-answer pairs (Gao et al., 2025a; Wang et al., 2025a; Zhuang et al., 2024; Zhang et al., 2024b; Guo et al., 2025a). Subsequent work has advanced the state

of the art with techniques like reinforcement learning (Yang et al., 2025; Wang et al., 2025b; Duan et al., 2025; Wei et al., 2025b). A parallel line of work instead focuses on strengthening the visual perception stage itself, either by decoupling diagram interpretation from linguistic reasoning or by promoting more deliberate, fine-grained parsing of geometric figures (Chen et al., 2025a; Guo et al., 2025b; Wei et al., 2025a). However, these models remain fundamentally text-centric: while they effectively interpret visual information in the input, they largely neglect vision as an active, generative component of the reasoning process itself.

**Visual Chain-of-Thought.** Unlike various textual chain-of-thought (Wei et al., 2023; Fang et al., 2025a,b), visual chain-of-thought aims to bridge this gap by integrating the generation of visual aids directly into the reasoning process. Existing approaches follow two main lines. The first leverages external tools, such as vision models to extract image details (Shao et al., 2024a; Chen et al., 2025b; Hu et al., 2024; OpenAI, 2025b; Gao et al., 2025b) or code interpreters to add auxiliary structures (Hu et al., 2024; Wang et al., 2025c,d). This approach, however, is constrained, as these tools are either non-generative or lack general applicability due to rigidity. The second line explores intrinsic VCoT, where models natively generate visual thoughts as an integral part of their reasoning (Cheng et al.,

2025; Li et al., 2025b,c; Chern et al., 2025; Jiang et al., 2025; Li et al., 2025a). Despite its promise, this approach has so far been demonstrated primarily in simpler domains like spatial games and struggles to produce the precise, logically consistent diagrams required for complex mathematical reasoning.

**Datasets and Benchmarks for Multimodal Mathematical Reasoning.** The progress in visual-mathematical reasoning is largely driven by the evolution of its benchmarks. While foundational datasets like Geometry3K (Lu et al., 2021) and ScienceQA (Lu et al., 2022) established the task, recent challenging benchmarks such as MMMU (Yue et al., 2024), MathVista (Lu et al., 2024), Mathvision (Wang et al., 2024), and MathVerse (Zhang et al., 2024a), among others (Qiao et al., 2024; Wang et al., 2025e; Sun et al., 2024), have pushed the limits of LMMs’ visual reasoning. However, a fundamental limitation persists: these benchmarks consist of static problem-solution pairs and lack the step-by-step *visual* demonstrations required to train models for dynamic, process-oriented reasoning. This is precisely the gap our work addresses with the introduction of MathCanvas-Instruct and the MathCanvas-Bench benchmark.

### 3 Method

In this section, we detail the methodology behind MathCanvas. We first describe the construction of our large-scale training corpora for visual manipulation and strategic reasoning (3.1). We then introduce MathCanvas-Bench, a dedicated benchmark for rigorous evaluation (3.2). Finally, we present our two-stage training recipe that leverages these resources to instill intrinsic VCoT capabilities in a unified LMM (3.3).

#### 3.1 Training Corpora Construction

##### 3.1.1 Million-scale Pretraining Corpus

To endow unified LMMs with the foundational visual synthesis and editing capabilities required for mathematical reasoning, we construct a comprehensive million-scale pretraining corpus comprising two complementary components: MathCanvas-Edit for diagram editing and MathCanvas-Imagen for diagram generation. The overall construction pipeline is shown in Figure 2.

**MathCanvas-Edit** is designed to teach models how to iteratively modify mathematical diagrams

through step-by-step transformations. We construct this dataset through a hybrid pipeline that combines complex competition-level geometry problems with systematically generated simple geometric figures, yielding a total of 5.2M edit trajectories.

**Competition-Level Mining.** We start with 128 geometry problems from mathematical competitions to serve as realistic seed configurations. Using these seeds, we employ the AlphaGeometry LLM (Trinh et al., 2024) with beam search to generate numerous auxiliary line drawing methods for each problem. We then filter for geometrically invalid constructions and render the corresponding diagram sequences, where each step is an edit operation (e.g., adding an auxiliary line, marking an angle). This iterative process yields 4.2M edit trajectories capturing the complexity of competition-level reasoning. To ensure visual diversity from this limited set of seeds, the rendering of each trajectory is controlled by a unique random seed, varying visual attributes like orientation and line styles.

**Foundational Structure Generation.** While competition problems provide realism, they tend toward complexity that may not adequately cover fundamental editing operations. To address this, we construct a complementary set of simple geometric figures using AlphaGeometry’s formal language. We first define a basic geometric primitive set (e.g., points, lines, circles) and a geometric relation set (e.g., circumcenter, incenter, parallel), the full details of which are provided in Appendix B.1. Then we develop an automated algorithm that randomly and incrementally adds geometric primitives and relations to these basic structures, creating progressively more complex diagrams. Invalid or degenerate configurations are filtered out through geometric constraint checking. By leveraging different random seeds during rendering, we obtain 1M additional edit trajectories that provide systematic coverage of fundamental geometric operations after three iterations of this synthetic generation process.

**MathCanvas-Imagen** focuses on teaching models to generate mathematical diagrams from textual descriptions. We construct it by aggregating and processing data from three complementary sources, resulting in 10M caption-to-diagram pairs.

**Re-purposing from MathCanvas-Edit.** We first leverage the edit trajectories in MathCanvas-Edit, extracting caption-to-diagram pairs from each editing step. After deduplication based on visual and textual similarity, we obtain 5.4M diverse caption-

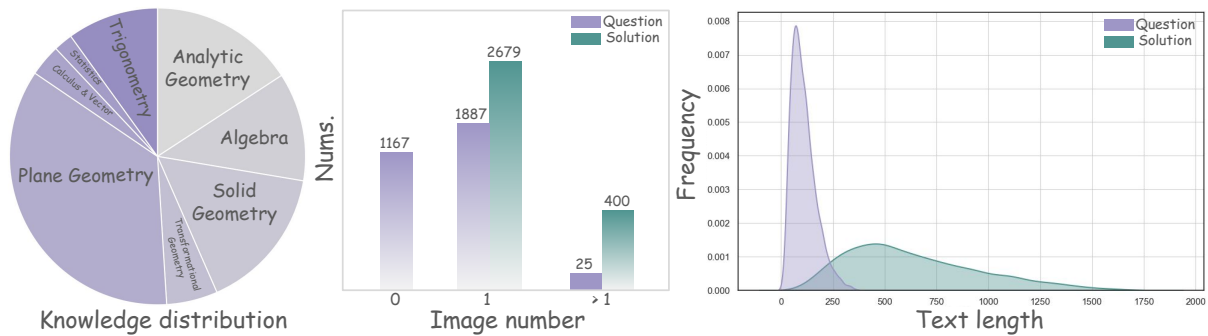


Figure 3: Statistical analysis of the MathCanvas-Bench test set. **Left:** Knowledge types distribution. **Middle:** Distribution of questions and solutions containing varying numbers of images. **Right:** Text length distribution of questions and solutions (measured in text tokens).

to-diagram pairs that inherently align with the types of diagrams needed for mathematical reasoning.

**Augmenting with Code-derived Captions.** To further scale our dataset, we utilize the ImgCode-8.6M (Wang et al., 2025a) dataset, which contains programmatically generated mathematical diagrams paired with source code. We first apply quality filtering to remove corrupted or low-quality images. We then employ GPT-4.1-mini to generate natural language captions by taking image-code pairs as input, producing descriptions that capture both the visual content and mathematical semantics of each diagram. This process yields 4M high-quality caption-to-diagram pairs with rich, descriptive captions with diverse mathematical diagrams.

**Incorporating Public Datasets.** Finally, we incorporate 612K caption-to-diagram pairs from existing public resources, including MAVIS (Zhang et al., 2024b) and TR-CoT (Deng et al., 2025b), which provide additional diversity in caption styles and diagram types, complementing our dataset.

Through this comprehensive construction process, the pretraining corpus provides a robust foundation for pretraining models on both diagram generation and editing, establishing the essential visual capabilities needed for intrinsic VCoT in mathematical reasoning.

### 3.1.2 MathCanvas-Instruct

To equip models with the ability to strategically interleave visual synthesis and editing actions with their textual reasoning process, we introduce MathCanvas-Instruct, the first large-scale dataset specifically designed for interleaved visual-textual mathematical reasoning.

**Dataset Construction** We begin by gathering 632K multimodal mathematics problems and so-

lutions from a wide array of middle school and high school textbooks, exams, and websites. These problems natively follow a multi-part sub-question structure (e.g., parts (1), (2), (3)), which we preserve throughout our pipeline without further splitting or merging. From this initial pool, we implement a rigorous multi-stage filtering pipeline to ensure data quality and relevance. First, we employ GPT-5 to analyze the problems, filtering out examples where the provided images served no role in the reasoning process. This step also standardized all mathematical formulas into LaTeX format, resulting in a refined set of 367K problems. A second round of filtering, also powered by GPT-5, removes problems that contained errors, lacked explicit answers, featured low-quality or unclear images, or consisted solely of drawing tasks. This left us with 303K high-quality problems.

To ensure the novelty and diversity of the dataset, we then perform both text and image deduplication, which yielded 222K unique problem-solution pairs. The images in the remaining dataset underwent a quality enhancement step using a super-resolution model, SwinIR (Liang et al., 2021), to improve clarity and detail before being resized to a uniform 512x512 resolution. Finally, GPT-4.1 is used to classify all problems into a hierarchical taxonomy of 8 major categories and fine-grained subcategories. This collection is then partitioned to form our evaluation benchmark, MathCanvas-Bench, with the remaining 219K examples constituting the MathCanvas-Instruct training set. Further statistics and examples for MathCanvas-Instruct are presented in Appendix B.2.

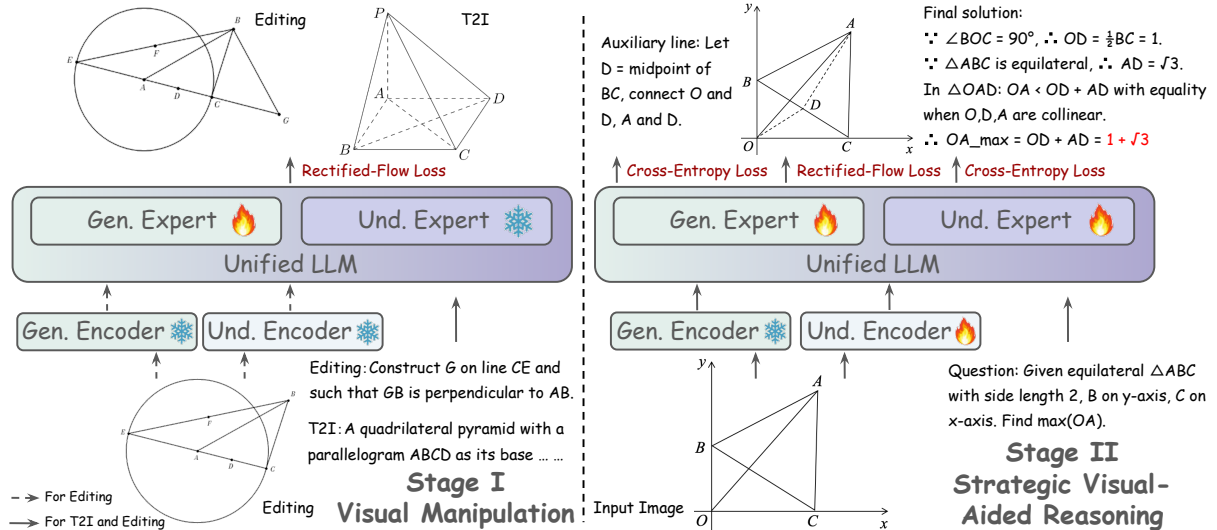


Figure 4: The two-stage training recipe for MathCanvas. **(Left) Stage I: Visual Manipulation.** The model’s Generation Expert is pretrained on our MathCanvas-Edit and MathCanvas-Imagen corpora to instill foundational diagram generation and editing skills. **(Right) Stage II: Strategic Visual-Aided Reasoning.** The entire model is then fine-tuned on MathCanvas-Instruct to learn the strategic interleaving of visual actions with textual reasoning.

### 3.2 The MathCanvas-Bench Evaluation Benchmark

**Benchmark Construction** We construct MathCanvas-Bench by sampling 3K problems from the 222K-pair collection described in Section 3.1.2. The construction process involves three key steps. First, we exclude all multiple-choice questions to ensure that evaluation relies on generative reasoning rather than random guessing. Second, to create a balanced test set, we perform weighted sampling across problem categories, setting the sampling weight for each category to the 0.7 exponential power of its proportion. This strategy increases the representation of less common problem types. Finally, to prevent data leakage, we remove any question from the remaining 219K training set that has a 5-gram Jaccard similarity score higher than 0.4 with any problem in MathCanvas-Bench. This process helps ensure a fair evaluation of model generalization. Further statistics on the final benchmark are shown in Figure 3.

**Evaluation Protocol** Our evaluation protocol relies on GPT-4.1 to ensure consistent and scalable assessment. For each problem, GPT-4.1 is tasked with extracting the final answers for every sub-question from the model’s output and comparing them against the ground-truth answers. The specific prompt templates used for this process are detailed in Appendix C. We employ two distinct

metrics to score performance:

**Complete Accuracy:** A binary score is awarded. A model receives 1 point only if the answers to all sub-questions are correct, and 0 otherwise. This metric evaluates the model’s ability to solve a problem completely.

**Weighted Scoring:** To provide a more granular evaluation of partial progress, this metric assigns exponentially increasing weights to each sub-question. The precise formula for this weighting scheme is detailed in Appendix C. The final score is the sum of the weights of the correctly answered sub-questions, a method that allows us to assess the model’s accuracy on intermediate steps within the reasoning chain.

Thus, MathCanvas-Bench provides a rigorous and challenging testbed for evaluating interleaved image-textual reasoning capabilities.

### 3.3 Two-Stage Training Recipe

We implement our framework on BAGEL (Deng et al., 2025a), a state-of-the-art unified LMM. Its architecture features two distinct transformer experts—one for understanding and one for generation—integrated within a single, unified model structure. This design provides a strong foundation for our approach. Our MathCanvas recipe enhances this architecture through a two-stage process, as illustrated in Figure 4: a foundational Stage I: Visual Manipulation, followed by Stage II: Strategic Visual-Aided Reasoning.

Model	Size	Think	Overall		Algebra	Analytic Geom.	Calc & Vector	Plane Geom.	Solid Geom.	Stats.	Transf. Geom.	Trig.
			Complete	Weighted								
<i>Closed-source (unified) LMMs</i>												
Gemini-2.5-Pro	-	✓	47.9	58.2	68.0	59.2	60.2	54.8	48.7	64.5	58.5	69.9
Gemini-2.5-Flash	-	✓	39.3	49.5	63.2	56.5	54.6	40.7	40.7	61.1	46.8	64.6
Gemini-2.0-Flash	-	✗	21.2	32.6	39.1	32.6	38.9	31.1	25.6	51.4	28.1	38.0
GPT-4.1	-	✗	19.0	30.0	40.4	30.7	37.1	24.1	25.1	54.0	21.5	42.5
GPT-4.1-mini	-	✗	14.6	26.3	35.7	30.5	36.5	22.0	22.4	24.8	19.7	30.3
GPT-4o	-	✗	9.9	19.4	21.6	17.7	21.8	19.5	18.6	17.4	13.2	23.0
GPT-5	-	✓	43.5	51.4	68.7	55.5	64.2	45.6	36.1	64.5	42.7	66.5
Claude-Sonnet-4	-	✓	25.0	37.8	44.8	38.9	49.3	33.8	33.0	46.9	30.3	47.6
Seed-1.6-Thinking	-	✓	44.1	55.2	67.7	57.5	55.9	52.2	45.0	65.1	56.8	60.7
Qwen3-VL-Plus	-	✓	40.9	51.5	67.0	54.6	56.9	45.9	42.0	66.7	49.3	58.9
Nano-Banana	-	✗	33.2	43.7	55.4	50.2	51.8	34.5	36.6	56.7	39.4	60.4
<i>Open-source (unified) LMMs</i>												
Qwen-2.5-VL-7B	7B	✗	8.9	18.7	19.5	19.0	19.2	20.6	18.7	10.7	13.9	15.0
Qwen-2.5-VL-32B	32B	✗	15.4	27.6	29.8	27.4	27.8	27.4	27.2	27.9	20.1	30.5
Qwen-2.5-VL-72B	72B	✗	21.1	32.8	30.6	19.5	36.4	34.5	33.5	23.9	33.6	48.9
Gemma-3-27b-it	27B	✗	15.8	26.6	31.3	28.4	34.4	25.8	21.0	40.0	21.0	26.9
InternVL3.5-8B	8B	✗	16.7	26.4	32.3	33.8	33.8	24.2	26.9	43.7	16.2	14.9
InternVL3.5-30B-A3B	30B	✗	11.7	22.2	22.2	19.9	15.1	24.9	24.3	22.1	17.4	18.4
Keye-VL-1.5-8B	8B	✓	17.1	27.0	33.1	28.0	26.2	27.0	23.6	29.5	20.9	26.3
BAGEL	7B	✗	8.3	18.5	18.1	13.1	17.1	20.8	23.0	10.9	19.4	13.3
BAGEL-Zebra-CoT	7B	✗	8.0	16.6	18.0	15.1	15.6	18.0	16.8	20.8	11.1	14.1
<b>BAGEL-Canvas</b>	7B	✗	21.9	34.4	29.9	27.2	17.9	40.0	35.3	23.2	29.3	40.4
△ Over Base Model			+13.6	+15.9	+11.8	+14.1	+0.8	+19.2	+12.3	+12.3	+9.9	+27.1

Table 1: Comparison of model performances across all mathematical subjects. The best **closed-source** and **open-source** highest accuracy of LMMs are marked in red and blue, respectively.

**Stage I: Visual Manipulation** The goal of this foundational stage is to instill robust visual synthesis and editing skills for mathematical diagrams. We pretrain the model on a mixture of our 5.2M-trajectory MathCanvas-Edit and 10M-pair MathCanvas-Imagen datasets. To foster iterative editing capabilities, each editing trajectory is structured as a continuous sequence of 2-4 diagram transformations. To preserve the model’s inherent reasoning abilities, we freeze the entire understanding pathway and exclusively train the Generation Expert via a Rectified-Flow Loss (Liu et al., 2022) on the diagram generation task (Figure 4, Stage I). This approach builds a strong visual foundation without catastrophic forgetting of its core understanding capabilities.

### Stage II: Strategic Visual-Aided Reasoning

With the visual foundation established, Stage II fine-tunes the model to intelligently interleave its drawing and reasoning faculties using our interleaved image-text dataset, MathCanvas-Instruct. To enable the model to strategically decide *when* to draw, it is trained on a token prediction task. Following each text segment (marked by the `<im_end>` token), the model must predict whether to generate the `<|vision_start|>` token to initiate a drawing, or the `<|endoftext|>` token to conclude the

response.

Importantly, this autoregressive token-level mechanism naturally supports **multi-step and iterative visual reasoning** within a single solution: after any text segment, the model can autonomously choose to generate a new diagram, update a previously drawn one, or terminate. This is reflected in our training data (Table 6, average 1.18 and up to 5 images per solution) and in the MathCanvas-Bench distribution (Figure 3, where 13% of problems involve multi-image solutions).

To inform *how* the model draws and understands, we process input and output images differently. All images provided in the question are encoded into clean VAE and ViT tokens, serving as visual context. For images within the solution, which the model must generate, we additionally include noised VAE tokens to compute the Rectified-Flow Loss. Unlike Stage I, all model components are unfrozen and trained jointly (Figure 4, Stage II). To enhance generation quality, we also leverage the architecture’s inherent dual Classifier-Free Guidance mechanism during inference. This orchestration stage culminates in a model that can autonomously generate diagrams as intermediate steps to solve complex problems. Detailed training hyperparameters are provided in Appendix A.

## 4 Experiments

We compare BAGEL-Canvas against 20 prominent LMMs, including top-performing proprietary models such as the Gemini series (2.5-Pro, 2.5-Flash, Nano-Banana, 2.0-Flash) (Comanici et al., 2025), the GPT series (GPT-5, GPT-4.1, GPT-4.1-mini, GPT-4o) (OpenAI, 2025a; OpenAI et al., 2024c,a), Claude-Sonnet-4 (Anthropic, 2025), other strong multimodal models like Seed-1.6-Thinking (Seed et al., 2025) and Qwen3-VL-Plus (Bai et al., 2025), and powerful open-source models, including the Qwen-2.5-VL series (7B, 32B, 72B) (Bai et al., 2025), Gemma-3-27b-it (Team et al., 2025), InternVL3.5 (8B, 30B) (Wang et al., 2025b), and Keye-VL-1.5-8B (Yang et al., 2025). We also include our base model, BAGEL (Deng et al., 2025a), and a variant, BAGEL-Zebra-CoT (Li et al., 2025a), to precisely measure the gains from our framework. All LMM evaluations are conducted using VLMEvalKit (Duan et al., 2024) to ensure a fair comparison. The comprehensive results are shown in Table 1.

### 4.1 Benchmark Results

As presented in Table 1, BAGEL-Canvas achieves a weighted score of 34.4% on our benchmark, establishing it as the top-performing open-source model. It surpasses all open-source competitors, including significantly larger models like Qwen-2.5-VL-72B (32.8) and InternVL3.5-30B-A3B (22.2). This result represents a substantial +15.9 point improvement over its base model, BAGEL, demonstrating the profound effectiveness of our training paradigm in unlocking advanced reasoning capabilities. Furthermore, BAGEL-Canvas proves to be highly competitive with proprietary systems, outperforming several prominent models such as Gemini-2.0-Flash (32.6) and GPT-4.1 (30.0).

An analysis of performance across mathematical domains reveals that BAGEL-Canvas exhibits the most significant gains in geometry-heavy subjects: Trigonometry (+27.1), Plane Geometry (+19.2), and Solid Geometry (+12.3). This result strongly supports our hypothesis that visual reasoning is particularly beneficial for geometric problem-solving. The model also shows substantial improvements in Analytic Geometry (+14.1) and Algebra (+11.8), suggesting that the ability to visualize functions and coordinate systems enhances reasoning in broader mathematical contexts. The modest gain in Calculus & Vector (+0.8) indicates that this domain

may require specialized reasoning capabilities beyond the scope of our current visual augmentation techniques.

### 4.2 Performance on Other Math Benchmarks

To assess the generalization capabilities of BAGEL-Canvas, we evaluate it on three established public benchmarks: the GPS category from MathVista’s testmini set (Lu et al., 2024), the full test set of MathVision (Wang et al., 2024), and the Text Dominant/Lite subsets from MathVerse’s testmini (Zhang et al., 2024a). As detailed in Table 2, BAGEL-Canvas demonstrates substantial and consistent improvements over its base model, BAGEL, across all benchmarks, with particularly strong gains on MathVerse (+17.9) and MathVista (+10.5). The detailed breakdown on MathVision further reveals significant improvements in subjects that benefit from visual intuition, such as Analytic Geometry (+22.6), Algebra (+13.0), and various plane geometry tasks (Angle: +17.3). Crucially, since these benchmarks require text-only solutions, this strong performance validates that our training paradigm fundamentally enhances the model’s intrinsic reasoning abilities, allowing it to generalize effectively to traditional problem-solving formats.

### 4.3 Ablation Studies

We conduct a series of ablation studies to dissect the contributions of the key components within our framework: the pretraining corpus and the role of the visual modality in the final reasoning stage.

**Effectiveness of the Pre-training Corpus.** We investigate the impact of our two-stage pre-training strategy by ablating the MathCanvas-Edit and MathCanvas-Imagen corpora. As shown in Table 3, removing the MathCanvas-Edit data (w/o MathCanvas-Edit) results in a 2.4-point drop in the weighted score. This highlights the importance of learning step-by-step diagram editing, a critical skill for solving complex problems that require constructing auxiliary elements. A further ablation, removing the entire pre-training stage (w/o MathCanvas-Imagen), leads to an additional 1.2-point performance decrease. This confirms that even foundational diagram generation capabilities provide a vital scaffold for the fine-tuning phase. Together, these results validate our two-stage pre-training approach, demonstrating that both generation and editing skills are essential for achieving optimal performance.

Model	MathVista	MathVerse		MathVision							
	(GPS)	(Text Dominant)	(Text Lite)	(test)	AnaG	Angle	Area	Len	SolG	Alg	Others
BAGEL	68.8	49.2	42.0	24.1	26.2	31.8	25.0	28.7	22.1	17.1	23.1
<b>BAGEL-Canvas</b>	79.3	65.4	59.9	32.9	48.8	49.1	35.2	37.9	31.2	30.1	27.9
$\Delta$	<b>+10.5</b>	<b>+16.2</b>	<b>+17.9</b>	<b>+8.8</b>	<b>+22.6</b>	<b>+17.3</b>	<b>+10.2</b>	<b>+9.2</b>	<b>+9.1</b>	<b>+13.0</b>	<b>+4.8</b>

Table 2: Generalization performance of BAGEL-Canvas compared to its base model (BAGEL) on three multimodal math benchmarks.  $\Delta$  indicates the absolute improvement. MathVision subject abbreviations: AnaG (Analytic Geometry), SolG (Solid Geometry), Alg (Algebra), Angle (Metric Geometry - Angle), Area (Metric Geometry - Area), and Len (Metric Geometry - Length).

Model	Overall	
	Complete	Weighted
<b>BAGEL-Canvas</b>	<b>21.9</b>	<b>34.4</b>
w/o MathCanvas-Edit	19.8	32.0
w/o MathCanvas-Imagen	18.2	30.8

Table 3: Ablation study on the pre-training corpora. We report the performance drop after removing the editing data (w/o MathCanvas-Edit) and the entire pre-training data (w/o MathCanvas-Imagen).

Model	Overall	
	Complete	Weighted
<b>BAGEL-Canvas</b>	<b>21.9</b>	<b>34.4</b>
– (Skip Image)	19.7	31.9
BAGEL-Canvas-Text	18.7	30.9

Table 4: Ablation study on the visual modality. BAGEL-Canvas-Text is a variant fine-tuned without any visual data. (– Skip Image) denotes the full model being constrained to text-only reasoning during inference.

### Importance of Visual Modality in Reasoning.

We analyze the importance of the visual modality through two ablations. First, we fine-tune a variant, BAGEL-Canvas-Text, using only the textual reasoning paths from MathCanvas-Instruct. Second, we constrain the full BAGEL-Canvas model to bypass visual generation during inference (– Skip Image). As shown in Table 4, both scenarios result in a significant performance drop. The BAGEL-Canvas-Text variant’s weighted score falls by 3.5 points, confirming that training on interleaved visual-textual data is essential for learning complex reasoning. Interestingly, the model that simply skips image generation at inference (– Skip Image) performs 1.0 point better than BAGEL-Canvas-Text, despite both producing text-only solutions. This suggests that our interleaved training paradigm not only teaches the model how to leverage visual aids but also fundamentally enhances its underlying textual reasoning capabilities.

## 5 Conclusion

We introduced MathCanvas, a comprehensive framework to endow Large Multimodal Models with intrinsic Visual Chain-of-Thought capabilities for mathematical reasoning. By leveraging our newly created large-scale datasets (MathCanvas-Edit, MathCanvas-Imagen, and MathCanvas-Instruct) in a two-stage training recipe, we taught our model, BAGEL-Canvas, to master diagram manipulation and strategically interleave it with textual deduction. This approach yielded an 86% relative improvement over strong baselines on our MathCanvas-Bench benchmark. Crucially, this training paradigm not only teaches the model *when* and *how* to draw, but also fundamentally enhances its core textual reasoning. Our work provides a robust foundation for future research into broader and more complex multimodal reasoning.

### Limitations

Despite its strong performance, our work has several limitations. First, the performance of BAGEL-Canvas is closely tied to the capabilities of its base model, BAGEL. A natural and immediate avenue for improvement is to apply our MathCanvas framework to more powerful, state-of-the-art open-source LMMs. Furthermore, our current training relies on supervised fine-tuning. Exploring advanced training paradigms, such as Reinforcement Learning (RL), could further enhance the model’s problem-solving strategies by directly rewarding logical correctness and the utility of the generated visual aids.

Second, our approach is primarily designed for middle and high school mathematics with a strong emphasis on geometry-intensive problems. Its effectiveness in more abstract domains such as advanced algebra, number theory, or topology remains less explored, as these fields often demand different and more abstract forms of visual rea-

soning (e.g., knot diagrams) that are beyond the capabilities of our current model.

Finally, our model’s intrinsic VCoT is limited to generating static images, a simplification of the dynamic and iterative nature of human visual thought. Furthermore, while our weighted scoring metric (Section 3.2) acknowledges partial progress on textual sub-questions, our evaluation protocol does not assess the fidelity of the intermediate visual manipulations. Future work could therefore explore more dynamic visual reasoning mechanisms and develop metrics to automatically assess the quality and logical soundness of the intermediate visual steps, enabling a more holistic analysis of visual-textual reasoning.

## Acknowledgements

This project is funded in part by Shenzhen Loop Area Institute, by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnoHK, in part by NSFC-RGC Project N\_CUHK498/24, and in part by Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515130008, XW).

## References

- Anthropic. 2025. [System card: Claude opus 4 & claude sonnet 4](#). Technical report, Anthropic. Accessed 2025-10-07.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Felix Chen, Hangjie Yuan, Yunqiu Xu, Tao Feng, Jun Cen, Pengwei Liu, Zeyang Huang, and Yi Yang. 2025a. [Mathflow: Enhancing the perceptual flow of mllms for visual mathematical problems](#). *Preprint*, arXiv:2503.16549.
- Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. 2025b. [Mint-cot: Enabling interleaved visual tokens in mathematical chain-of-thought reasoning](#). *arXiv preprint arXiv:2506.05331*.
- Zihui Cheng, Qiguang Chen, Xiao Xu, Jiaqi Wang, Weiyun Wang, Hao Fei, Yidong Wang, Alex Jinpeng Wang, Zhi Chen, Wanxiang Che, and Libo Qin. 2025. [Visual thoughts: A unified perspective of understanding multimodal chain-of-thought](#). *Preprint*, arXiv:2505.15510.
- Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. 2025. [Thinking with generated images](#). *arXiv preprint arXiv:2505.22525*.
- Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasapat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. 2025a. [Emerging properties in unified multimodal pretraining](#). *Preprint*, arXiv:2505.14683.
- Linger Deng, Linghao Zhu, Yuliang Liu, Yu Wang, Qunyi Xie, Jingjing Wu, Gang Zhang, Yingying Zhu, and Xiang Bai. 2025b. [Theorem-validated reverse chain-of-thought problem generation for geometric reasoning](#). *Preprint*, arXiv:2410.17885.
- Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. 2025. [Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning](#). *arXiv preprint arXiv:2505.17022*.
- Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, and 1 others. 2024. [Vlmevalkit: An open-source toolkit for evaluating large multi-modality models](#). In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201.
- Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, and 1 others. 2025a. [Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing](#). *arXiv preprint arXiv:2503.10639*.
- Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. 2025b. [Flux-reason-6m & prism-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark](#). *arXiv preprint arXiv:2509.09680*.

- Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. 2025a. [G-llava: Solving geometric problem with multi-modal large language model](#). *Preprint*, arXiv:2312.11370.
- Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. 2025b. [Interleaved-modal chain-of-thought](#). *Preprint*, arXiv:2411.19488.
- Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhua Chen, and Xiang Yue. 2025a. [Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale](#). *Preprint*, arXiv:2412.05237.
- Zixian Guo, Ming Liu, Qilong Wang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. 2025b. [Integrating visual interpretation and linguistic reasoning for math problem solving](#). *Preprint*, arXiv:2505.17609.
- Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. [Visual sketchpad: Sketching as a visual chain of thought for multimodal language models](#). *Preprint*, arXiv:2406.09403.
- Dongzhi Jiang, Renrui Zhang, Haodong Li, Zhuofan Zong, Ziyu Guo, Jun He, Claire Guo, Junyan Ye, Rongyao Fang, Weijia Li, Rui Liu, and Hongsheng Li. 2025. [Draco: Draft as cot for text-to-image preview and rare concept generation](#). *Preprint*, arXiv:2512.05112.
- Ang Li, Charles Wang, Kaiyu Yue, Zikui Cai, Ollie Liu, Deqing Fu, Peng Guo, Wang Bill Zhu, Vatsal Sharan, Robin Jia, and 1 others. 2025a. [Zebra-cot: A dataset for interleaved vision language reasoning](#). *arXiv preprint arXiv:2507.16746*.
- Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025b. [Imagine while reasoning in space: Multimodal visualization-of-thought](#). *arXiv preprint arXiv:2501.07542*.
- Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025c. [Imagine while reasoning in space: Multimodal visualization-of-thought](#). *Preprint*, arXiv:2501.07542.
- Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. [Swinir: Image restoration using swin transformer](#). *Preprint*, arXiv:2108.10257.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. [Flow straight and fast: Learning to generate and transfer data with rectified flow](#). *Preprint*, arXiv:2209.03003.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). *Preprint*, arXiv:1711.05101.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. [Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts](#). In *International Conference on Learning Representations (ICLR)*.
- Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. [Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning](#). *Preprint*, arXiv:2105.04165.
- Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Taffjord, Peter Clark, and Ashwin Kalyan. 2022. [Learn to explain: Multimodal reasoning via thought chains for science question answering](#). In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024a. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024b. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.
- OpenAI. 2025a. [GPT-5 System Card](#). Technical report, OpenAI. Accessed on [YYYY-MM-DD].
- OpenAI. 2025b. [OpenAI o3 and o4-mini System Card](#). Technical report, OpenAI. Accessed on [YYYY-MM-DD].
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024c. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Yifan Zhang, Xiao Zong, Yida Xu, Muxi Diao, Zhimin Bao, Chen Li, and Honggang Zhang. 2024. [We-math: Does your large multimodal model achieve human-like mathematical reasoning?](#) *Preprint*, arXiv:2407.01284.
- Houxing Ren, Zimu Lu, Weikang Shi, Haotian Hou, Yunqiao Yang, Ke Wang, Aojun Zhou, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2025a. [Alignment with fill-in-the-middle for enhancing code generation](#). *Preprint*, arXiv:2508.19532.

- Houxing Ren, Mingjie Zhan, Zhongyuan Wu, Aojun Zhou, Junting Pan, and Hongsheng Li. 2025b. [Reflectioncoder: Learning from reflection sequence for enhanced one-off code generation](#). *Preprint*, arXiv:2405.17057.
- ByteDance Seed, :, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, Yufeng Yuan, Yu Yue, Lin Yan, Qiyang Yu, Xiaochen Zuo, Chi Zhang, Ruofei Zhu, Zhecheng An, and 255 others. 2025. [Seed1.5-thinking: Advancing superb reasoning models with reinforcement learning](#). *Preprint*, arXiv:2504.13914.
- Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. [Visual cot: Advancing multimodal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning](#). *Advances in Neural Information Processing Systems*, 37:8612–8642.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024b. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Kai Sun, Yushi Bai, Ji Qi, Lei Hou, and Juanzi Li. 2024. [Mm-math: Advancing multimodal math evaluation with process evaluation and fine-grained classification](#). *Preprint*, arXiv:2404.05091.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Trieu Trinh, Yuhuai Wu, Quoc Le, He He, and Thang Luong. 2024. [Solving olympiad geometry without human demonstrations](#). *Nature*.
- Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. [Measuring multimodal mathematical reasoning with math-vision dataset](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Ke Wang, Junting Pan, Linda Wei, Aojun Zhou, Weikang Shi, Zimu Lu, Han Xiao, Yunqiao Yang, Houxing Ren, Mingjie Zhan, and Hongsheng Li. 2025a. [Mathcoder-vl: Bridging vision and code for enhanced multimodal mathematical reasoning](#). *Preprint*, arXiv:2505.10557.
- Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023. [Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning](#). *Preprint*, arXiv:2310.03731.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025b. [Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency](#). *Preprint*, arXiv:2508.18265.
- Yikun Wang, Siyin Wang, Qinyuan Cheng, Zhaoye Fei, Liang Ding, Qipeng Guo, Dacheng Tao, and Xipeng Qiu. 2025c. [Visuothink: Empowering lvlm reasoning with multimodal tree search](#). *Preprint*, arXiv:2504.09130.
- Yikun Wang, Yibin Wang, Dianyi Wang, Zimian Peng, Qipeng Guo, Dacheng Tao, and Jiaqi Wang. 2025d. [Geometryzero: Improving geometry solving for llm with group contrastive policy optimization](#). *Preprint*, arXiv:2506.07160.
- Zhikai Wang, Jiashuo Sun, Wenqi Zhang, Zhiqiang Hu, Xin Li, Fan Wang, and Deli Zhao. 2025e. [Benchmarking multimodal mathematical reasoning with explicit visual dependency](#). *Preprint*, arXiv:2504.18589.
- Haoran Wei, Youyang Yin, Yumeng Li, Jia Wang, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. 2025a. [Slow perception: Let's perceive geometric figures step-by-step](#). *Preprint*, arXiv:2412.20631.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, arXiv:2201.11903.
- Lai Wei, Yuting Li, Kaipeng Zheng, Chen Wang, Yue Wang, Linghe Kong, Lichao Sun, and Weiran Huang. 2025b. [Advancing multimodal reasoning via reinforcement learning with cold start](#). *Preprint*, arXiv:2505.22334.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. [Qwen2.5-math technical report: Toward mathematical expert model via self-improvement](#). *Preprint*, arXiv:2409.12122.
- Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, Fan Yang, Guorui Zhou, Guowang Zhang, Han Shen, Hao Peng, Haojie Ding, Hao Wang, Haonan Fan, Hengrui Ju, and 42 others. 2025. [Kwai keye-vl 1.5 technical report](#). *Preprint*, arXiv:2509.01563.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and

3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. 2023. [Mammoth: Building math generalist models through hybrid instruction tuning](#). *Preprint*, arXiv:2309.05653.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and 1 others. 2024a. [Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?](#) *arXiv preprint arXiv:2403.14624*.

Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, Peng Gao, Chunyuan Li, and Hongsheng Li. 2024b. [Mavis: Mathematical visual instruction tuning with an automatic data engine](#). *Preprint*, arXiv:2407.08739.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2024. [Math-puma: Progressive upward multi-modal alignment to enhance mathematical reasoning](#). *Preprint*, arXiv:2408.08640.

## A Training Details

We implement our framework on top of the publicly available **BAGEL-7B-MoT** (Deng et al., 2025a) model. All training experiments were conducted on a cluster of 16 GPUs, each equipped with 80GB of VRAM. We use the AdamW (Loshchilov and Hutter, 2019) optimizer for both training stages. The detailed hyperparameters for our two-stage training recipe, corresponding to Stage I (Visual Manipulation) and Stage II (Strategic Visual-Aided Reasoning), are provided in Table 5.

**Stage I** In this stage, the primary objective is to train the model’s visual generation capabilities. As described in Section 3.3, we freeze the entire understanding expert and only train the generation expert. The loss is solely based on the Rectified-Flow objective (Liu et al., 2022) for diagram generation, hence the absence of a Cross-Entropy loss component. We employed a slightly higher ViT condition dropout rate (0.3) to regularize the model and prevent overfitting to the visual features of the pretraining data.

**Stage II** In the second stage, all model components are unfrozen to enable joint optimization. The model is trained on a combined loss function: a Cross-Entropy loss for predicting the next token (either text or the special  $\langle |vision\_start| \rangle$  and  $\langle |endof\text{text}| \rangle$  token), weighted by 0.25, and the Rectified-Flow loss for generating diagrams, weighted by 1.0. The learning rate is halved, and the number of training steps is reduced, which is typical for fine-tuning tasks. The ViT condition dropout is lowered to 0.1 to better leverage visual context during strategic reasoning.

## B Dataset Details

### B.1 MathCanvas-Edit and MathCanvas-Imagen

#### Details on Foundational Structure Generation.

As described in Section 3.1, the Foundational Structure Generation pipeline for the MathCanvas-Edit dataset relies on an automated algorithm that randomly and incrementally adds geometric primitives and relations. This section specifies the exact sets used in this process.

**Geometric Primitive Set.** The generation process initiates by selecting one of 18 basic geometric objects from the following set:

- segment, angle

- Triangles: triangle, iso\_triangle (isosceles), r\_triangle (right), triangle\_ab, ieq\_triangle (equilateral), risos (right isosceles)
- Quadrangles: rectangle, isquare, trapezoid, r\_trapezoid (right), eq\_trapezoid (isosceles), quadrangle, eq\_quadrangle (equilateral), eqdia\_quadrangle (equal-diagonal)
- Polygons: pentagon, eq\_pentagon (equilateral)

**Geometric Relation Set.** Subsequently, the algorithm iteratively applies relations from a predefined set of 41 constructions. These are categorized by the number of new points they introduce (typically one or two).

- **1-Point Relations (37):** angle\_bisector, angle\_mirror, circle, circumcenter, eq\_triangle, eqangle2, eqdistance, foot, incenter, excenter, intersection\_cc, on\_bline, intersection\_lc, on\_aline, intersection\_ll, on\_line, intersection\_lp, intersection\_lt, intersection\_pp, intersection\_tt, lc\_tangent, midpoint, mirror, nsquare, on\_bline, on\_circle, on\_pline, on\_tline, on\_dia, orthocenter, parallelogram, psquare, reflect, s\_angle, shift, on\_opline, eqangle3, on\_circum
- **2-Point Relations (4):** square, trisegment, trisect, tangent

The automated algorithm randomly samples from these sets to build progressively more complex diagrams, ensuring systematic coverage of fundamental geometric operations.

**Geometric Validity Filtering.** Our construction pipeline is built on top of AlphaGeometry’s formal geometric language with a Python-based engine for instantiation. For each newly proposed construction, the engine verifies (i) satisfiability of the geometric constraints, and (ii) whether the resulting configuration can be robustly rendered (i.e., all point coordinates are numerically stable and non-degenerate). We filter out any construction that triggers validity errors, such as InvalidQuadSolveError, InvalidLineIntersectError, or PointTooCloseError, which typically indicate degenerate or ill-posed configurations. When such an error occurs, the candidate is discarded, and constructions are resampled until a valid diagram sequence is obtained.

Hyperparameter	Stage I	Stage II
<b>Optimizer &amp; Scheduler</b>		
Learning Rate (LR)	$2 \times 10^{-5}$	$1 \times 10^{-5}$
LR Scheduler	Cosine Decay	Cosine Decay
Min Learning Rate	$1 \times 10^{-7}$	$1 \times 10^{-7}$
Warmup Steps	2,000	500
Total Training Steps	80,000	16,000
<b>Model &amp; Loss</b>		
EMA Decay Rate	0.999	0.995
Rectified-Flow Timestep Shift	2.0	2.0
Cross-Entropy (CE) Loss Weight	N/A	0.25
Rectified-Flow (MSE) Loss Weight	1.0 (Implicit)	1.0
Frozen Components	Understanding Expert	None
<b>Batching &amp; Tokenization</b>		
Max Tokens per Batch	46,080	51,200
Max Tokens per Sample	8,192	25,600
<b>Regularization (Dropout)</b>		
Text Condition Dropout	0.1	0.1
ViT Condition Dropout	0.3	0.1
VAE Condition Dropout	0.1	0.1

Table 5: Key hyperparameters for the two-stage training process. "N/A" indicates that the parameter was not applicable to that stage.

**Examples.** An example from the MathCanvas-Edit dataset is presented in Figure 6. Examples from the MathCanvas-Imagen dataset are shown in Figures 7 and 8.

## B.2 MathCanvas-Instruct

**Dataset Statistics.** We present the knowledge point distribution of the MathCanvas-Instruct training set in Figure 5. Table 6 demonstrates the statistical characteristics of the MathCanvas-Instruct dataset, comprising 219K problems, of which 65% are multimodal and 35% are text-only. We have also analyzed the distribution of problem sources, the length of questions and solutions, and the number of images they contain.

**Examples.** We showcase examples from the MathCanvas-Instruct dataset in Figures 9, 10, 11.

## C Benchmark Evaluation Details

### C.1 Weighted Scoring Weights

The weights for our Weighted Scoring metric are calculated using an exponential growth factor of 1.3, valuing later sub-questions more heavily. The specific formula for the weight  $w_i$  of the  $i$ -th sub-

question in a problem with  $N$  sub-questions is:

$$w_i = \frac{1.3^{i-1}}{\sum_{j=1}^N 1.3^{j-1}}$$

Since our benchmark contains problems with a maximum of four sub-questions, we use the following pre-calculated, normalized weights for evaluation. The final score for a problem is the sum of the weights of the correctly answered sub-questions.

- **For 2 sub-questions:** [0.4348, 0.5652]
- **For 3 sub-questions:** [0.2506, 0.3258, 0.4236]
- **For 4 sub-questions:** [0.1616, 0.2101, 0.2732, 0.3551]

### C.2 Evaluation Template

Tables 7 and 8 display the prompt templates used for MathCanvas-Bench evaluation.

## D Additional Qualitative Results

To further illustrate the limitations of even the most advanced LMMs when they lack intrinsic VCoT capabilities, we present qualitative examples of their performance on problems that benefit from visual

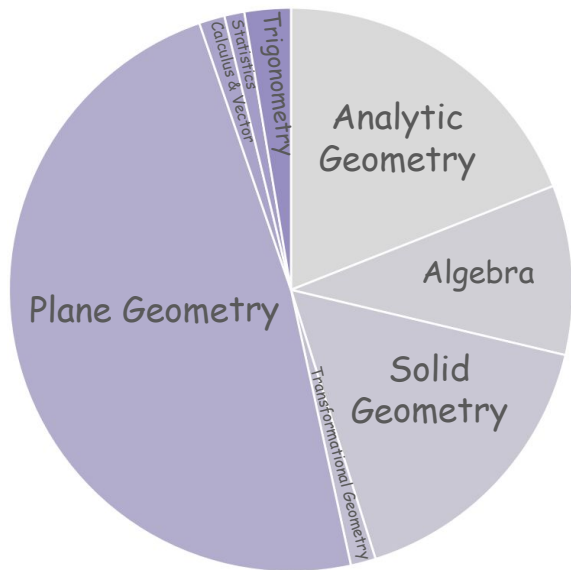


Figure 5: Distribution of knowledge type of MathCanvas-Instruct dataset.

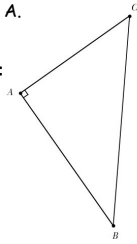
manipulation. Figure 12 shows the solutions from Gemini-2.5-Pro and GPT-5 for the problem featured in Figure 1 of the main paper, demonstrating their reliance on complex and sometimes flawed algebraic approaches. We provide more qualitative results of BAGEL-Zebra-CoT, Nano-Banana, and our method in Figure 13.

Statistics	Number
<b>Total Samples</b>	<b>218,604</b>
- Text questions	35%
- Multimodal questions	65%
- Middle school questions	<b>63%</b>
- Grade 7	6%
- Grade 8	17%
- Grade 9	77%
- High school questions	<b>37%</b>
- Grade 10	12%
- Grade 11	16%
- Grade 12	72%
- One question	68%
- Two sub-questions	18%
- Three sub-questions	12%
- Four or more sub-questions	2%
<b>Question length (text tokens)</b>	
- Maximum	466
- Average	107.92
<b>Solution length (text tokens)</b>	
- Maximum	2001
- Average	539.66
<b>Multimodal Question Image</b>	
- Maximum number	5
- Average number	1.03
<b>Solution Image</b>	
- Maximum number	5
- Average number	1.18

Table 6: More statistics of MathCanvas-Instruct dataset.

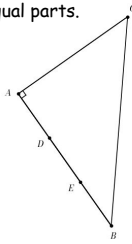
**Base caption:** Let  $ABC$  be a right triangle with the right angle at vertex  $A$ .

**Image:**



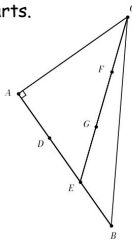
**Step 1:** Construct points  $D$  and  $E$  that divide segment  $AB$  into three equal parts.

**Image:**



**Step 2:** Construct points  $F$  and  $G$  that divide segment  $CE$  into three equal parts.

**Image:**



**Step 3:** Connect  $CD$ .

**Image:**

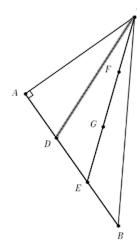
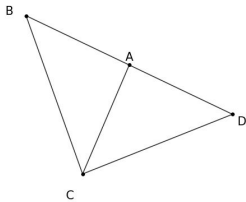


Figure 6: An example from MathCanvas-Edit dataset.

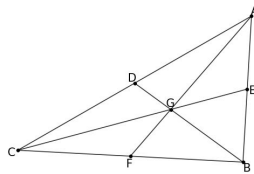
**Caption:** In triangle  $ABC$ ,  $\angle BAC$  is a right angle and  $\angle ABC$  and  $\angle ACB$  are equal. The shape  $D$  is the mirror image of  $B$  across the line  $CA$ .

**Image:**



**Caption:** Triangle  $CAB$  has  $G$  as its centroid, with  $D$ ,  $E$ , and  $F$  being the midpoints of sides  $CA$ ,  $BA$ , and  $CB$  respectively. The length of segment  $AB$  is half of segment  $AC$ .

**Image:**



**Caption:** Let  $ABC$  be a right isosceles triangle with the right angle at  $A$ . Construct points  $D$  and  $E$  that divide segment  $BC$  into three equal parts. Construct points  $F$  and  $G$  that divide segment  $BD$  into three equal parts. Construct  $H$  as the incenter of triangle  $ACD$ .

**Image:**

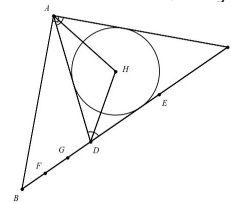
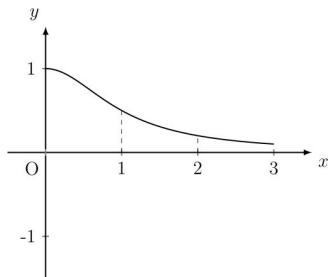


Figure 7: Examples from MathCanvas-Imagen dataset.

**Caption:** The image depicts a Cartesian coordinate system with the  $x$ -axis and  $y$ -axis labeled as  $x$  and  $y$ . The axes intersect at the origin, labeled as  $O$ . A smooth curve is plotted, representing the function  $y = 1 / (1 + x^2)$ . This curve starts at  $y = 1$  when  $x = 0$  and asymptotically approaches the  $x$ -axis as  $x$  increases. Two vertical dashed lines are drawn from the  $x$ -axis to the curve at  $x = 1$  and  $x = 2$ . These lines intersect the curve at points where the  $y$ -values are  $1/2$  and  $1/5$ , respectively. These intersections highlight the values of the function... illustrating its decreasing nature.

**Image:**



**Caption:** The image depicts a circle centered at point  $O$ , with three points  $A$ ,  $B$ , and  $C$  located on the circumference. The points are positioned with  $A$  in the upper left,  $B$  in the upper right, and  $C$  in the lower left. The line segments  $OA$ ,  $OB$ , and  $OC$  are radii of the circle. Additionally, the line segment  $AC$  is drawn, forming the triangle  $OAC$  with the radii  $OA$  and  $OC$ . The spatial arrangement suggests that triangle  $OAC$  is an isosceles triangle with  $OA = OC$ . The overall configuration highlights the geometric properties of the circle and the isosceles triangle within it.

**Image:**

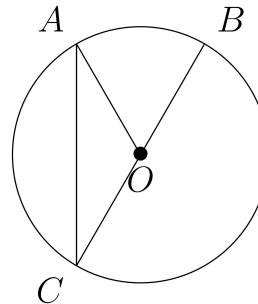


Figure 8: Examples from MathCanvas-Imagen dataset.

**Question:**

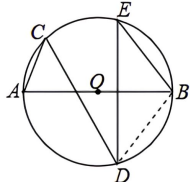
In the circle with center  $O$ ,  $AB$  is a diameter. Points  $C, D, E$  lie on the circle.  $DE$  is perpendicular to  $AB$ . Given  $\angle ABE=50^\circ$ , find the measure of  $\angle ACD$ .

Options: A.  $90^\circ$ , B.  $50^\circ$ , C.  $45^\circ$ , D.  $30^\circ$ .

**Solution:**

**Text:** Draw segment  $BD$ .

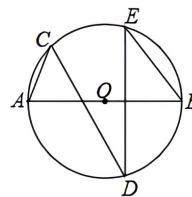
**Image:**



**Text:** Since  $AB$  is a diameter and  $DE$  is perpendicular to  $AB$  at its midpoint  $O$ , by the perpendicular diameter theorem,  $E$  and  $D$  are endpoints of a chord perpendicular to the diameter, so  $BE$  and  $BD$  are equal radii in isosceles triangle  $BDE$ , implying  $BE=BD$  and thus  $\angle ABD=\angle ABE$ . From  $\angle ABE=50^\circ$ , we get  $\angle ABD=50^\circ$ .  $\angle ACD$  and  $\angle ABD$  subtend the same arc  $AD$  of the circle, so by the inscribed angle theorem,  $\angle ACD=\angle ABD=50^\circ$ . Therefore, the correct option is B.

Figure 9: An example from MathCanvas-Instruct dataset.

**Input image:**



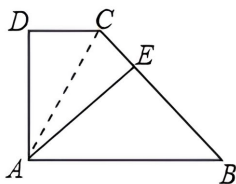
**Question:**

In right trapezoid  $ABCD$ ,  $AB \parallel CD$ ,  $AD \perp CD$ . Given  $AB=BC$ , and  $AE \perp BC$  at  $E$ . Determine the relationship between  $CD$  and  $CE$ .

**Solution:**

**Text:** Draw segment  $AC$ .

**Image:**



**Text:** Since  $AB=BC$ , triangle  $ABC$  is isosceles with vertex at  $B$ , hence  $\angle BAC=\angle ACB$ . Because  $AB \parallel CD$ , corresponding angles give  $\angle ACD=\angle ACB=\angle BAC$ . Therefore  $\angle ACD=\angle ACE$ . Given  $AD \perp CD$  and  $AE \perp BC$ , we have  $\angle ADC=\angle AEC=90^\circ$ . In triangles  $ACD$  and  $ACE$ :  $\angle ADC=\angle AEC$ ,  $\angle ACD=\angle ACE$ , and  $AC$  is common. By AAS, triangle  $ACD \cong ACE$ . Corresponding sides in congruent triangles are equal, so  $CD=CE$ . Thus,  $CD=CE$ .

**Input image:**

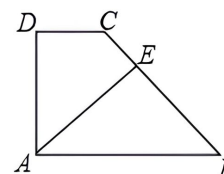


Figure 10: An example from MathCanvas-Instruct dataset.

**Question:**

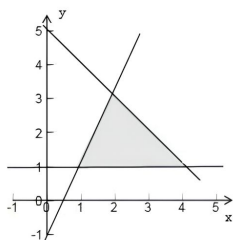
Real numbers  $x, y$  satisfy the constraints:  $y \geq 1$ ,  $y \leq 3x - 1$ ,  $x + y \leq m$ . If the minimum value of the objective function  $z = 5x - 4y$  is  $-3$ , then  $m =$  ( )

Options: A. 3 B. 2 C. 4 D.  $11/3$

**Solution:**

**Text:** The feasible region is defined by the inequalities  $y \geq 1$ ,  $y \leq 3x - 1$ , and  $x + y \leq m$ . Draw the feasible region.

**Image:**



**Text:** The minimum value of the objective function  $z = 5x - 4y$  will occur at one of the vertices of this region. Based on the slopes of the boundary lines and the objective function, the minimum value is achieved at the intersection of the lines  $y = 3x - 1$  and  $x + y = m$ . First, find the coordinates of this vertex by solving the system  $y = 3x - 1$  and  $x + y = m$ . This gives  $x = (m+1)/4$  and  $y = (3m-1)/4$ . Next, substitute these coordinates into the objective function:  $z = 5x - 4y = 5((m+1)/4) - 4((3m-1)/4) = (-7m + 9)/4$ . We are given that the minimum value of  $z$  is  $-3$ . So, we set  $(-7m + 9)/4 = -3$ . Solving for  $m$  gives  $-7m + 9 = -12$ , which leads to  $-7m = -21$ , so  $m = 3$ . The correct option is A.

Figure 11: An example from MathCanvas-Instruct dataset.

You are an expert mathematics teacher and a precise data evaluator. Your task is to analyze a given math problem, compare a predicted solution against a ground truth answer, and determine if the prediction is correct.

**INPUT FORMAT:**

You will be provided with a JSON string containing the following fields:

- `question_text`: The full text of the mathematical problem.
- `ground_truth_answer`: The correct, final answer text. This is the gold standard and is already extracted.
- `prediction_solution`: The full solution text from the model, from which you must extract the final answer(s).

**TASK & OUTPUT REQUIREMENTS:**

Your output must be a single, valid JSON object. The process involves two main steps: **Answer Parsing and Extraction** and **Correctness Judgment**.

**Step 1: Answer Parsing and Extraction**

Your first task is to create two lists of answers: `gt_answers` and `pred_answers`. The structure of the `gt_answers` list defines the required structure for the `pred_answers` list.

**1.1 Parsing ground\_truth\_answer:**

- The `ground_truth_answer` is a clean, final answer.
- Your task is to **parse** it into a list (`gt_answers`).
- **CRITICAL PARSING RULE:** The **only** condition for creating a list with multiple elements is the presence of explicit multi-part answer tags (e.g., `<1>...</1>`, `<2>...</2>`).
- **If tags are present:** Extract the content of each tag into a separate list element. *Example:* `"<1>5 cm</1><2>10 cm</2>"` becomes `["5 cm", "10 cm"]`.
- **If no such tags are present:** The **entire, unmodified string** must be treated as the **single element** of the list. Do not split the string by characters, words, commas, or any other pattern. *Example 1:* `"ABC"` must become `["ABC"]`, **NOT** `["A", "B", "C"]`. *Example 2:* `"x=5, y=10"` must become `["x=5, y=10"]`, **NOT** `["x=5", "y=10"]`.
- The `gt_answers` list will never contain null elements and its length defines the number of sub-questions.

**1.2 Extracting from prediction\_solution:**

- Your primary task is to **extract** the final answer(s) from the `prediction_solution` text to create the `pred_answers` list.
- **IMPORTANT:** The answers to different sub-questions may appear in different places within the `prediction_solution`, not necessarily grouped together at the end. You must treat this as a **matching task**.

Table 7: The prompt template (part 1) used by GPT-4.1 for mathematical reasoning evaluation.

- For each part of the `gt_answers` list, you must scan the **entire** `prediction_solution` to find the corresponding predicted answer. Look for explicit labels (e.g., "(1)", "Part A"), final conclusions, boxed answers (e.g., `\boxed{...}`), or statements that directly answer a part of the original question.
- The final `pred_answers` list **must have the exact same length as the `gt_answers` list**.
- For each sub-question, if you cannot find a corresponding answer in the `prediction_solution`, you **must** use `null` as a placeholder in that position. **This rule is critical and applies in all cases where an answer is missing, including when the `prediction_solution` appears incomplete or is truncated before all sub-questions are addressed.**

**CRITICAL RULE:** The final `gt_answers` and `pred_answers` lists **must** be of equal length. The number of parts in the `ground_truth_answer` dictates the required length for both lists.

### Step 2: Correctness Judgment

Your second task is to compare the `pred_answers` list against the `gt_answers` list, element by element.

#### Judgment Rules:

- **Numerical Equivalence:** Treat numbers as correct if they are mathematically equivalent (e.g., 5, 5.0, 10/2). Allow for minor floating-point rounding differences.
- **Textual Equivalence:** For text answers, judge based on semantic meaning, not exact matching. Ignore case, whitespace, and phrasing differences (e.g., "CB is parallel to PD" is equivalent to "Line CB || Line PD").
- **Generate Correctness List:** Create a boolean list named `correctness`. The *i*-th element is `true` if the *i*-th predicted answer is correct, `false` otherwise. This list **must** have the same length as the answer lists.

#### Final JSON Output Structure:

Your entire response must be a single, valid JSON object matching the schema below. Do not include any text outside of this JSON object.

```
{
  "analysis": "A brief step-by-step explanation...",
  "gt_answers": [
    "string",
    ...
  ],
  "pred_answers": [
    "string or null",
    ...
  ],
  "correctness": [
    true/false,
    ...
  ]
}
```

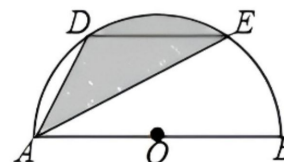
**INPUT DATA:** `{input_data}`

Table 8: The prompt template (part 2) used by GPT-4.1 for mathematical reasoning evaluation. The text highlighted in cyan is replaced with the specific input data for each problem being evaluated.

**Question:**

In the semicircle with diameter  $AB$ , let  $O$  be the midpoint of  $AB$ . Find the area of the shaded region given  $AB = 2$ ,  $\text{arc } AD = \text{arc } DE = \text{arc } EB$ .

**Input image:**



**Gemini-2.5-Pro:**

1. Determine Angles and Radius: Since the diameter  $AB = 2$ , the radius  $R = 1$ . The  $180^\circ$  arc of the semicircle is divided into three equal parts, so  $\text{arc } AD = \text{arc } DE = \text{arc } EB = 60^\circ$ . The corresponding central angle  $\angle DOE$  is also  $60^\circ$ .
2. Use Properties of Parallel Lines: Connect  $OD$  and  $OE$ . Since  $OD = OE = 1$  (radii) and  $\angle DOE = 60^\circ$ ,  $\triangle DOE$  is an equilateral triangle. This implies that the line segment  $DE$  is parallel to the diameter  $AB$ .
3. Equate and Calculate Area: The shaded triangle,  $\triangle ADE$ , and the triangle  $\triangle ODE$  share the same base  $DE$ . Because their vertices  $A$  and  $O$  lie on a line ( $AB$ ) that is parallel to the base  $DE$ , the triangles have equal heights. Therefore,  $\text{Area}(\triangle ADE) = \text{Area}(\triangle ODE)$ .

The area of the equilateral triangle  $\triangle ODE$  with side length 1 is:  
 $\text{Area} = (\text{side}^2 \times \sqrt{3}) / 4 = (1^2 \times \sqrt{3}) / 4 = \sqrt{3} / 4$ .

**GPT-5:**

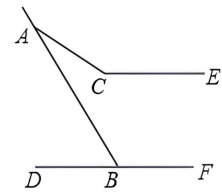
1. Establish Coordinates: Let the semicircle's center  $O$  be the origin  $(0, 0)$ . With diameter  $AB = 2$ , the radius is  $r = 1$ . Place the diameter on the x-axis, so  $A = (-1, 0)$  and  $B = (1, 0)$ .
2. Determine Vertex Coordinates: The  $180^\circ$  arc is divided into three equal  $60^\circ$  arcs. The angles for the vertices from the positive x-axis are  $E$  at  $60^\circ$ ,  $D$  at  $120^\circ$ , and  $A$  at  $180^\circ$ . Their coordinates  $(r \cdot \cos \theta, r \cdot \sin \theta)$  are:
  - \*  $A = (-1, 0)$
  - \*  $D = (\cos 120^\circ, \sin 120^\circ) = (-1/2, \sqrt{3}/2)$
  - \*  $E = (\cos 60^\circ, \sin 60^\circ) = (1/2, \sqrt{3}/2)$
3. Calculate Area: The area of  $\triangle ADE$  can be found using the shoelace formula (vector cross product). Define two vectors originating from vertex  $A$ :  $AD = D - A = (1/2, \sqrt{3}/2)$  and  $AE = E - A = (3/2, \sqrt{3}/2)$ . The area calculation is as follows:  
 $\text{Area} = 1/2 | (AD_x)(AE_y) - (AD_y)(AE_x) | = 1/2 | (1/2)(\sqrt{3}/2) - (\sqrt{3}/2)(3/2) | = 1/2 | \sqrt{3}/4 - 3\sqrt{3}/4 | = 1/2 | -2\sqrt{3}/4 | = \sqrt{3}/4$ .

Figure 12: Leading LMMs (Gemini-2.5-Pro and GPT-5) solving the problem from Figure 1 via text-only reasoning. Their complex or incorrect algebraic approaches, in contrast to the elegant geometric solutions unlocked by intrinsic VCoT, demonstrate the necessity of our MathCanvas framework.

**Question:**

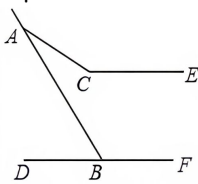
Given  $CE \parallel DF$ ,  $\angle ABF=100^\circ$ , and  $\angle CAB=20^\circ$ , find the measure of  $\angle ACE$ .

Input image:



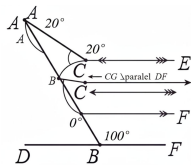
**BAGEL-Zebra-CoT:**

The problem asks for the measure of  $\angle ACE$ . We are given that  $CE \parallel DF$ . This suggests that we can use properties of parallel lines and transversals to find the required angle. We are also given  $\angle ABF=100^\circ$  and  $\angle CAB=20^\circ$ . Let's visualize the problem to better understand the relationships between the angles.



The diagram shows the lines  $CE$  and  $DF$  are parallel. The line segment  $AB$  intersects  $DF$  at point  $B$ . We are given  $\angle ABF=100^\circ$ . Since  $AB$  is a transversal cutting the parallel lines  $CE$  and  $DF$ , we can use the property that the sum of the interior angles on the same side of a transversal is  $180^\circ$ . Therefore,  $\angle ABF + \angle CBA = 180^\circ$ . We can calculate  $\angle CBA$  as follows:  $\angle CBA = 180^\circ - \angle ABF = 180^\circ - 100^\circ = 80^\circ$ . Next, we need to find  $\angle ACE$ ... The final answer is  $\angle ACE = 80^\circ$ .

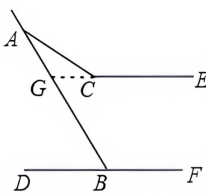
**Nano-Banana:**



To find the measure of  $\angle ACE$ , we can draw an auxiliary line through point  $C$  parallel to  $DF$  (and therefore also parallel to  $CE$ ). Let this line be  $CG$ , with  $G$  to the right of  $C$ . Since  $CG \parallel DF$ , we have:  $\angle GCB = \angle CBD$  (alternate interior angles). Since  $DF$  is a straight line,  $\angle ABC + \angle ABF = 180^\circ$ . So,  $\angle ABC = 180^\circ - \angle ABF = 180^\circ - 100^\circ = 80^\circ$ ... The final answer is  $60^\circ$ .

**Ours:**

Extend  $EC$  to meet line  $AB$  at point  $G$ .



Since  $CE \parallel DF$ ,  $\angle AGE = \angle ABF = 100^\circ$ . In triangle  $ACG$ , the exterior angle at  $C$  equals the sum of the two remote interior angles:  $\angle ACE = \angle CAB + \angle AGE = 20^\circ + 100^\circ = 120^\circ$ . Therefore, answer is  $120^\circ$ .

Figure 13: Comparison of BAGEL-Zebra-CoT, Nano-Banana, and our method.