



# Figure It Out: Improve the Frontier of Reasoning with Executable Visual States

Meiqi Chen, Fandong Meng<sup>†</sup>, Jie Zhou  
WeChat AI, Tencent Inc

{meiqiichen, fandongmeng, withtomzhou}@tencent.com

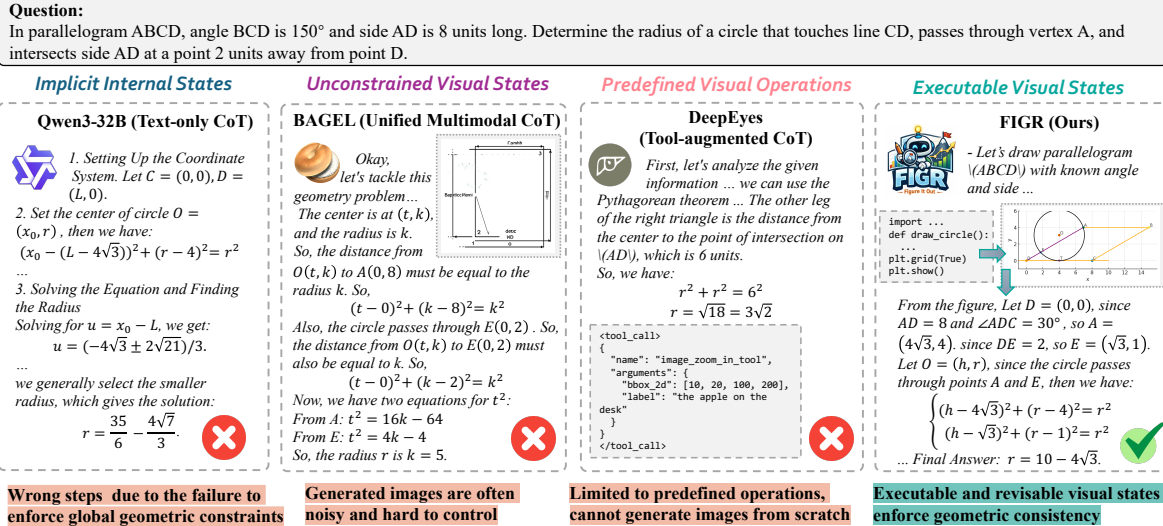


Figure 1: Comparison of reasoning paradigms via intermediate states. Text-only reasoning relies on implicit symbolic states; unified multimodal models generate less controllable visual states; tool-augmented models operate on given images with predefined transformations. In contrast, **FIGR** constructs executable visual states *within* the reasoning loop, enabling precise, revisable diagrams that help enforce geometric consistency.

## Abstract

Complex reasoning problems often involve implicit spatial and geometric relationships that are not explicitly encoded in text. While recent reasoning models perform well across many domains, purely text-based reasoning struggles to capture structural constraints in complex settings. In this paper, we introduce **FIGR**, which integrates executable visual construction into multi-turn reasoning via end-to-end reinforcement learning. Rather than relying solely on textual chains of thought, **FIGR** externalizes intermediate hypotheses by generating executable code that constructs diagrams within the reasoning loop. An adaptive reward mechanism selectively regulates when visual construction is invoked, enabling more consistent reasoning over latent global properties that are difficult to infer from text alone. Experiments on seven challenging mathematical benchmarks demonstrate that **FIGR** outperforms strong text-only chain-of-thought baselines, improving the base model by 13.12% on AIME 2025 and 11.00% on BeyondAIME. These results highlight the effectiveness of precise, controllable figure construction of **FIGR** in enhancing complex reasoning ability.

## 1 Introduction

Human reasoning is deeply intertwined with the ability to externalize information into visual form. When faced with complex constraints or multi-step relationships, people naturally draw diagrams, sketch intermediate states, or construct spatial layouts to reduce cognitive load and clarify underlying structure (Tversky et al., 2019; Norman, 2024). Cognitive science further suggests that such diagrammatic reasoning allows people to discover patterns that are difficult or even impossible to glean from text alone (Donald, 1993; Mandler, 2010).

Figure 1 provides an illustrative comparison of how different reasoning paradigms behave on a geometry problem involving angle constraints, tangency conditions, and point intersections. In text-only chain-of-thought models (CoT) (Wei et al., 2022), all spatial relations must be implicitly maintained through symbolic expressions. This places a heavy burden on internal representations and can lead to cascading algebraic errors when subtle geometric constraints are misinterpreted. A natural

\*<https://github.com/chenmeiqii/FIGR>.

<sup>†</sup>Corresponding author.

extension is to consider unified multimodal models (Team, 2024; Wu et al., 2025b; Deng et al., 2025; Li et al., 2025a), which generate images as part of the reasoning process to make spatial relationships explicit. While conceptually appealing, these models lack precise control over the generated visual content. Since image generation is not grounded in executable constraints, even small spatial inconsistencies may propagate across reasoning steps, limiting their reliability in problems that require fine-grained geometric precision.

In parallel, tool-augmented large vision-language models (LVLMs) incorporate external visual tools or predefined APIs to assist reasoning, spanning prompt engineering (Gupta and Kembhavi, 2023; Hu et al., 2024; Surís et al., 2023), fine-tuning (Wu and Xie, 2024; Wang et al., 2025), and reinforcement learning (Zheng et al., 2025; Zhang et al., 2025). By delegating visual operations to explicit tools, these approaches improve controllability and execution fidelity. However, they are typically constrained to operating on *given* images and predefined transformations (e.g., zooming or cropping), which limits their ability to autonomously construct task-specific diagrams required by complex reasoning problems.

Motivated by these limitations, we introduce **FIGR**, which integrates visual construction directly into the reasoning loop via end-to-end reinforcement learning. Rather than relying on unconstrained image generation or fixed toolsets, **FIGR** produces executable code that bridges symbolic reasoning and visual rendering. This allows the model to actively construct and iteratively refine diagrams during multi-turn inference, using rendered figures as explicit, stateful feedback—analogueous to how humans repeatedly sketch and update diagrams when solving complex problems. As illustrated in Figure 1, this mechanism helps enforce geometric consistency and supports correct reasoning in cases where other paradigms fail.

To regulate this reasoning–rendering process, we design an *Adaptive Reward Mechanism* that selectively encourages visual construction when it is beneficial, while discouraging unnecessary or spurious rendering behaviors. Importantly, this design removes the need for a supervised visual reasoning cold-start stage. Instead, **FIGR** can be initialized from an instruction-tuned model (e.g., Qwen3-VL-32B-Instruct (Bai et al., 2025a)) and refined purely through reinforcement learning. Through this process, **FIGR** autonomously learns

both when to invoke visual reasoning and how to integrate execution feedback into its reasoning trajectory, without relying on task-specific supervised visual reasoning data.

Overall, our main contributions are as follows:

- We propose **FIGR**, which actively performs visual thinking during multi-turn reasoning.
- We introduce an adaptive reward mechanism that regulates the selective use of visual reasoning without requiring supervised cold-start.
- We demonstrate substantial improvements over the base model and text-only RL baselines across diverse mathematical reasoning benchmarks.

## 2 Related Work

### 2.1 Text-based and Programmatic Reasoning

A series of works extends chain-of-thought (CoT) prompting (Wei et al., 2022) by introducing structured reasoning strategies for large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023). Program-of-Thought (PoT) (Chen et al., 2022) and Chain-of-Code (Li et al., 2024) interleave natural language with executable code, allowing models to offload arithmetic and symbolic manipulation to external interpreters. Tree-of-Thought (ToT) (Yao et al., 2023) and its variants further represent reasoning as a branching search process, enabling exploration of alternative solution paths and pruning of unpromising branches. More recently, reinforcement learning has been applied to regulate reasoning behaviors, such as learning when to expand or truncate reasoning trajectories (Guo et al., 2025), or when and how to invoke a code interpreter during problem solving (Feng et al., 2025).

Despite these advances, such approaches operate primarily over textual or symbolic representations. Intermediate reasoning states remain implicit and must be maintained internally by the model, which poses persistent challenges for tasks involving geometry, kinematics, or other forms of spatial reasoning. Without explicit external representations, complex relational constraints are prone to the accumulation of errors, limiting the reliability of purely text-based and programmatic reasoning in structurally demanding settings.

### 2.2 Unconstrained Visual Generation for Reasoning

Large vision-language models (LVLMs) (Liu et al., 2023; Bai et al., 2025b) demonstrate that LLMs can

interpret images through relatively shallow alignment layers, but most of them remain limited to text-only outputs. Recent unified multimodal architectures (Team, 2024; Wu et al., 2025b; Deng et al., 2025; Li et al., 2025a) further extend generation to both text and images within a single model, enabling visual content to be produced as part of the reasoning process.

While conceptually appealing, these approaches rely on unconstrained image generation without explicit mechanisms to enforce geometric precision or relational consistency. As a result, the generated visual representations can be noisy or spatially imprecise, limiting their effectiveness as reliable intermediate states for reasoning. This limitation becomes particularly pronounced in mathematical and scientific domains, where even minor visual inaccuracies can propagate across reasoning steps and lead to significant downstream errors.

### 2.3 Tool-Augmented Visual Reasoning

Another line of research augments LVLMs with external visual tools, rendering engines, or predefined APIs. These methods decompose a problem into sub-steps, apply visual operators such as zooming, cropping, or drawing auxiliary lines, and interpret the results to guide subsequent reasoning (Gupta and Kembhavi, 2023; Surís et al., 2023; Hu et al., 2024; Shao et al., 2024a; Wu and Xie, 2024; Li et al., 2025b; Shen et al., 2025; Wang et al., 2025). Reinforcement learning-based pipelines further improve tool usage efficiency by explicitly rewarding appropriate invocation patterns (Zheng et al., 2025; Zhang et al., 2025).

Tool-augmented visual reasoning approaches provide strong controllability, as their visual operations are deterministic and explicitly defined. However, they are typically restricted to operating on given images and a fixed set of predefined transformations. Consequently, they lack the ability to autonomously construct new diagrams, geometric configurations, or abstract visual representations that are not present in the input. This restriction limits their applicability in reasoning tasks that require dynamic generation and refinement of task-specific visual structures.

## 3 Methodology

In this section, we describe the main components of **FIGR**. Section 3.1 introduces the executable reasoning loop, Section 3.2 details the reinforcement

learning procedure, and Section 3.3 presents the adaptive reward mechanism.

### 3.1 Executable Visual Construction within the Reasoning Loop

**FIGR** integrates executable visual construction directly into its multi-turn reasoning process. For each input, the model interleaves textual inference and code generation to construct diagrams, analogous to how humans iteratively sketch intermediate states while reasoning.

Formally, at each reasoning step  $t$ , the model maintains a state consisting of a textual context  $h_t$  (including prior reasoning steps, code outputs, and interpreter feedback) and an optional rendered diagram  $I_t$ . The policy  $\pi_\theta$  samples an action:

$$a_t \sim \pi_\theta(\cdot | h_t, I_t), \quad (1)$$

where  $a_t$  is either (i) a textual continuation or (ii) a code snippet  $c_t$  generated by the policy itself. When a code action is emitted, it is executed by a sandboxed interpreter, which may produce textual feedback  $T_{t+1}$  and/or a rendered diagram  $I_{t+1}$ :

$$(T_{t+1}, I_{t+1}) = \text{Interpreter}(c_t). \quad (2)$$

The textual context is then updated accordingly:

$$h_{t+1} = \text{UpdateContext}(h_t, c_t, T_{t+1}). \quad (3)$$

This process repeats until a termination token is emitted or a step limit is reached. By representing diagrams as executable and reproducible states, **FIGR** enables precise and controllable visual feedback throughout the reasoning process.

### 3.2 Reinforcement Learning with Multi-turn Executable Reasoning

Training **FIGR** requires learning long-horizon reasoning policies that interleave textual inference with executable visual construction. In this setting, the quality of individual actions (e.g., whether to emit code at a given step) cannot be evaluated locally, as their utility depends on the final reasoning outcome and the resulting visual states. We therefore formulate learning at the trajectory level and employ reinforcement learning to optimize multi-turn reasoning behaviors.

We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024b), which is well-suited for trajectory-level supervision with delayed terminal rewards. For each input question, GRPO

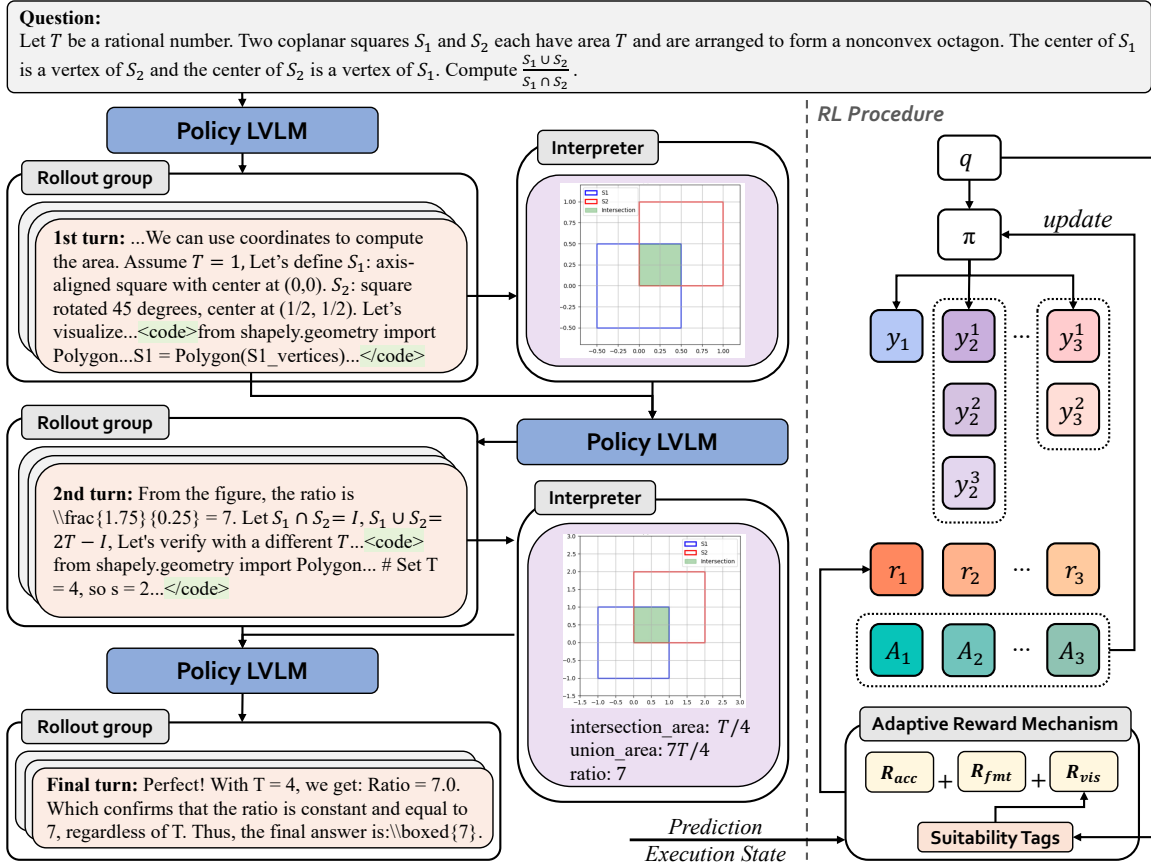


Figure 2: Overview of **FIGR**. **FIGR** alternates between textual reasoning and *executable visual construction* within a unified reasoning loop. An adaptive reward mechanism selectively regulates when visual construction is invoked, based on question suitability, execution outcomes, and final answer correctness.

samples a group of  $G$  candidate trajectories under the current policy. Each trajectory  $\tau_i$  consists of a sequence of textual and code actions, together with their executed interpreter feedback, and terminates when the model emits a special end token. After rollout termination, we compute a scalar reward:

$$R_i = R_{\text{acc}}(\tau_i) + R_{\text{fmt}}(\tau_i) + R_{\text{vis}}(\tau_i), \quad (4)$$

which jointly reflects final answer correctness, output format compliance, and the appropriateness of visual construction (in Section 3.3). Let  $\hat{y}(\tau_i)$  be the answer extracted from the final `<answer>` block, and let  $\text{fmt}(\tau_i)$  indicate whether the trajectory satisfies the required output format. We define:

$$R_{\text{acc}}(\tau_i) = \mathbb{I}[\hat{y}(\tau_i) = y], \quad (5)$$

$$R_{\text{fmt}}(\tau_i) = \mathbb{I}[\text{fmt}(\tau_i) = 1]. \quad (6)$$

The format mandates well-formed `<think>`, `<answer>`, and, when applicable, `<tool_call>` spans. Importantly, rewards are assigned only at the trajectory level, as intermediate actions contribute to reasoning quality through their cumulative effects.

To stabilize learning under such delayed supervision, GRPO compares trajectories generated from the same prompt and uses the group-average reward as a baseline:

$$\bar{R} = \frac{1}{G} \sum_{i=1}^G R_i. \quad (7)$$

The relative advantage of each trajectory is then defined as:

$$\hat{A}_i = R_i - \bar{R}, \quad (8)$$

which provides a low-variance learning signal that emphasizes relative behavioral differences, such as when executable visual construction leads to better outcomes than purely textual reasoning.

The policy is updated by maximizing the following surrogate objective:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}, \{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}} \frac{1}{G} \sum_{i=1}^G \frac{1}{|\tau_i|} \sum_{t=1}^{|\tau_i|} M_{i,t}(\theta) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}), \quad (9)$$

where

$$M_{i,t}(\theta) = \min \left[ r_{i,t}(\theta) \hat{A}_i, \text{clip}(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_i \right]. \quad (10)$$

Here,  $r_{i,t}(\theta) = \frac{\pi_{\theta}(a_{i,t}|s_{i,t})}{\pi_{\theta_{\text{old}}}(a_{i,t}|s_{i,t})}$  denotes the probability ratio between the updated and previous policies,  $\varepsilon$  is the clipping threshold, and  $\pi_{\text{ref}}$  is a reference policy used for KL regularization. This objective encourages the policy to increase the likelihood of trajectories that achieve higher relative rewards, while maintaining stability during optimization.

Overall, GRPO enables **FIGR** to learn multi-turn reasoning strategies in which the decision to invoke visual construction and the resulting reasoning quality are optimized jointly, without requiring step-wise supervision or an explicit value model.

### 3.3 Adaptive Reward Mechanism

While executable visual construction can significantly benefit certain reasoning problems, indiscriminate invocation of diagram generation may introduce unnecessary complexity or distract from effective textual reasoning. To address this trade-off, we introduce an adaptive reward mechanism that acts as a *policy-level control signal*, guiding when visual construction should be invoked during multi-turn reasoning.

For each input question, we first estimate whether diagrammatic reasoning is likely to be beneficial by querying an auxiliary language model classifier, which outputs a binary suitability label  $s \in \{0, 1\}$ . Importantly, this classifier does not provide step-wise supervision; instead, it modulates reward magnitude, encouraging selective use of visual construction and making the policy robust to moderate misclassification.

During reinforcement learning, the visual-invocation reward  $R_{\text{vis}}$  is evaluated only at the trajectory level and is coupled with final answer correctness. Specifically, a positive visual reward is granted to correct trajectories that successfully invoke executable visual construction, with a higher weight assigned when diagrammatic reasoning is predicted to be appropriate for the task:

$$R_{\text{vis}} = \begin{cases} 1.0, & \text{if } y = \hat{y}, s = 1, \text{exec} = 1, \\ 0.2, & \text{if } y = \hat{y}, s = 0, \text{exec} = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

Here,  $y$  and  $\hat{y}$  denote the ground-truth and predicted final answers, respectively, and `exec` indicates successful execution of the generated code.

Rather than explicitly supervising individual drawing actions, this reward formulation biases the policy toward reasoning strategies in which visual construction is used purposefully and effectively. By tying visual rewards to overall task success, the model learns to treat diagram construction as a meaningful intermediate state rather than a mandatory or habitual operation. This reduces the incentive for redundant or spurious visual actions while preserving the flexibility to exploit executable visual feedback when it improves reasoning.

## 4 Experiments

### 4.1 Experimental Setup

**Implementation Details** We conduct reinforcement learning using the VeRL framework (Sheng et al., 2024), with Qwen3-VL-32B-Instruct (Bai et al., 2025a) as the base policy. We set the maximum number of interaction rounds to 3, the maximum generation length to 32,768 tokens, and the sampling temperature to 0.7. We initialize from an instruction-tuned model to ensure stable structured outputs (e.g., executable code and tool-call delimiters) during multi-turn RL, as it yields more reliable formatting in early training.

**Training Dataset** We train **FIGR** on DeepMath-103K (He et al., 2025), a rigorously decontaminated dataset of 103,000 challenging mathematical problems with verifiable answers. Following Section 3.3, we annotate each instance with a binary suitability label indicating whether executable visual construction is likely to be beneficial. An auxiliary language model (Deepseek-V3 (Liu et al., 2024)) is used for this annotation. We perform eight independent annotation runs and apply majority voting. Prompt details are provided in Appendix B.1.

**Evaluation Datasets** To ensure comparability with prior work, we conduct evaluations on a suite of mathematical reasoning datasets that are widely used as standard benchmarks. We use AIME 2024 (AIME, 2024), AIME 2025 (AIME, 2025), BeyondAIME (Seed et al., 2025), MATH 500 (Hendrycks et al., 2021; Lightman et al., 2024), AMC (AMC, 2023), MinervaMath (Lewkowycz et al., 2022), and the *OE\_TO\_maths\_en\_COMP* subset of OlympiadBench (He et al., 2024). The dataset details are introduced in Appendix A.

**Baselines** We compare **FIGR** with the following competitive baselines:

Model	AIME 2024	AIME 2025	Beyond AIME	MATH 500	AMC	Minerva Math	Olymp Bench	Avg.
<i>Large Language Models</i>								
Qwen3-235B-A22B (Thinking)	83.80	80.78	52.00	95.40	93.98	46.69	74.04	75.24
Qwen3-32B (Non-Thinking)	31.00	20.20	16.00	88.60	61.45	39.34	52.08	44.10
Qwen3-32B (Thinking)	81.40	72.90	40.00	97.30	93.98	45.22	71.96	71.82
<i>Large Vision-Language Models</i>								
GLM-4.5V (108B)	76.20	67.34	47.00	96.20	87.95	38.60	70.03	69.62
Qwen3-VL-8B-Instruct	61.46	46.20	30.00	94.20	81.93	40.07	67.21	60.15
Qwen3-VL-32B-Instruct	73.33	66.20	43.00	93.60	84.34	41.18	69.73	67.34
Text-only RL	73.33	69.22	46.00	94.40	87.95	43.38	70.33	69.23
<b>FIGR (ours)</b>	<b>79.58</b>	<b>79.32</b>	<b>54.00</b>	<b>95.00</b>	<b>93.98</b>	<b>44.49</b>	<b>72.40</b>	<b>74.11</b>
<b>Gains</b>	<b>+6.25</b>	<b>+13.12</b>	<b>+11.00</b>	<b>+1.40</b>	<b>+9.64</b>	<b>+3.31</b>	<b>+2.67</b>	<b>+6.77</b>

Table 1: Main results (%) of **FIGR** on seven mathematical reasoning benchmarks compared with several competitive baselines. **Green-colored** font indicates improvement over the baseline Qwen3-VL-32B-Instruct.

- **Implicit-State Reasoning Models.** These models generate text-only outputs that implicitly maintain all intermediate states. We include three LLMs: Qwen3-235B-A22B (Thinking) (Yang et al., 2025) and Qwen3-32B (Non-Thinking, Thinking) (Yang et al., 2025); and three LVLMs: GLM-4.5V (108B), Qwen3-VL-8B-Instruct, and Qwen3-VL-32B-Instruct.
- **Unconstrained Visual Generation Models.** Unified multimodal models (UMMs) that generate images as part of the reasoning process, including Bagel-7B-MoT (Deng et al., 2025) and Bagel-Zebra-CoT (Li et al., 2025a).
- **Tool-Augmented Vision-Language Models (TAVLMs).** Models that rely on predefined visual tools or APIs operating on given images, including DeepEyes (Zheng et al., 2025) and Chain-of-Focus (Zhang et al., 2025).

We additionally include a text-only RL baseline trained with GRPO on DeepMath-103K, which optimizes trajectory-level correctness and format rewards without executable visual construction or interpreter feedback. For all baseline models, we set the maximum generation length, sampling temperature, and other hyperparameters following the recommended configurations from their respective papers or reports. The prompt templates are shown in Appendix B.2. Due to space constraints, for UMMs and TAVLMs, we report results on a subset of datasets in the main ablation study for clarity; full results are deferred to Appendix C.

**Evaluation Metrics** To ensure a stable evaluation, we default to pass@ $k$  evaluation (Chen, 2021)

and report pass@1 metrics. Specifically, we generate 64 responses for each question of AIME 2024 and AIME 2025 (i.e.,  $k = 64$ ), and  $k = 1$  for the remaining datasets. The pass@1 metric is then computed as:  $\text{pass@1} = \frac{1}{k} \sum_{i=1}^k p_i$ , where  $p_i$  denotes the correctness of the  $i$ -th response.

## 4.2 Main Results

As shown in Table 1, **FIGR** achieves substantial gains by learning to integrate executable visual construction into multi-turn reasoning. Across seven mathematical benchmarks, **FIGR** achieves an average accuracy of 74.11%, exceeding the base policy (Qwen3-VL-32B-Instruct) by 6.77% and the text-only RL baseline by 4.88%.

Performance gains are particularly pronounced on challenging benchmarks such as AIME 2025 and BeyondAIME, where reasoning often requires maintaining global geometric or structural constraints. Notably, **FIGR** outperforms both the equivalently sized text-only reasoning models (e.g., Qwen3-32B Thinking) and the larger LVLM GLM-4.5V on average. These results suggest that the gains come from learning a policy that selectively constructs controllable, executable visual states as intermediate representations during multi-turn reasoning.

## 4.3 Ablation Study

As shown in Table 2, we conduct ablation studies on two representative datasets: AIME 2025 and BeyondAIME, to evaluate the contributions of different components in **FIGR**.

**Prompt Engineering (PE) on the Base Model.** Prompt engineering introduces structured multimodal reasoning patterns without updating model

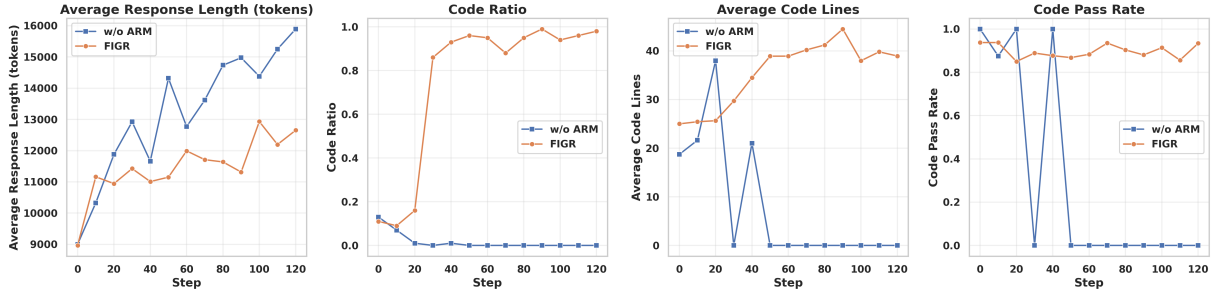


Figure 3: Ablation analysis across training steps. We track response length, code ratio, code lines, and code pass rate under different settings. **FIGR** exhibits sustained and structured executable visual construction with stable execution behavior, whereas removing the adaptive reward mechanism causes code usage to collapse during training.

parameters. While this leads to moderate performance gains on some datasets, the absence of a learning signal prevents the model from internalizing when and how such behaviors should be applied. As a result, the induced drawing behaviors remain unstable, which can limit the attainable performance gains.

Model	AIME 2025	Beyond AIME
Base Model	66.20	43.00
+ PE	63.33	46.00
+ SFT	53.33	38.00
+ Text-only RL	73.33	46.00
+ Bagel generated img	64.32	41.00
+ Qwen generated img	62.93	41.00
Bagel-7B-MoT	10.00	1.00
DeepEyes	2.34	0.00
<b>FIGR (ours)</b>	<b>79.32</b>	<b>54.00</b>
w/o ARM	70.00	49.00
w/o visual feedback	76.67	47.00

Table 2: Ablation Study on **FIGR**. The base model is Qwen3-VL-32B-Instruct, “PE” denotes prompt engineering, “SFT” denotes supervised fine-tuning, and “ARM” denotes the adaptive reward mechanism.

### Performance Comparison between SFT and RL.

We compare supervised fine-tuning (SFT) and reinforcement learning (RL) under the same data budget. Specifically, SFT is performed on the DeepMath-103K dataset using the provided distilled solutions generated by DeepSeek-R1 (Guo et al., 2025). Despite the strong teacher model, SFT leads to a noticeable performance degradation on challenging benchmarks (e.g., on AIME 2025, accuracy drops from 66.20% to 53.33%). In contrast, text-only reinforcement learning generally improves performance across datasets. This suggests that, in this setting, SFT may overfit to training trajectories and generalize poorly to unseen problems,

whereas outcome-driven RL enables more robust and transferable problem-solving behavior.

### Injecting Visual Information without Active Visual Reasoning.

We evaluate two baselines that provide *passively generated diagrams* as static context (+Bagel img and +Qwen img). These diagrams are generated from the problem text using Bagel-7B-MoT and Qwen-Image (Wu et al., 2025a), respectively, but are not constructed or revised during reasoning. Although these baselines expose the model to visual content, they do not consistently improve performance over the base model or text-only RL baseline. These results show that passively injected visual inputs, without executable construction or feedback integration, fail to serve as reliable intermediate states. Without a closed reasoning-execution loop, such images neither enforce global constraints nor provide actionable feedback for subsequent reasoning steps.

### Performance of UMMs and TAVLMs.

Both unified multimodal models and tool-augmented vision-language models exhibit consistently weak performance. Unified models are vulnerable to cascading errors when tackling highly complex problems. Tool-augmented methods are restricted to predefined operations on given images and cannot flexibly construct task-specific diagrams. Consequently, neither paradigm provides reliable and informative visual feedback for complex mathematical reasoning. Due to space limitations, their full results are deferred to Appendix C.

### Ablations on FIGR Components.

We further design two ablations to examine the contributions of **FIGR**’s components: (1) Removing the adaptive reward mechanism (ARM). We disable the visual-invocation reward, while keeping all other components unchanged. This ablation removes the

**Question:** Define  $f(x) = \sin^6 x + \cos^6 x + k \left| \sin^4 x + \cos^4 x \right|$  for some real number  $k$ . Determine all real numbers  $k$  for which there exists a real number  $c$  such that  $f(c) = 0$ .

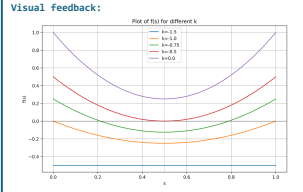
Qwen3-VL-32B-Instruct	w/o visual feedback	FIGR
<p>Let <math>t = \sin^2 x \in [0, 1]</math>, <math>1 - t = \cos^2 x</math>. Then <math>\sin^4 x = t^2</math>, <math>\cos^4 x = (1-t)^2</math>, <math>\sin^6 x = t^3</math>, <math>\cos^6 x = (1-t)^3</math>. Thus <math>f(x) = g(t)</math> where <math>g(t) = t^3 + (1-t)^3 + k \left  t^2 + (1-t)^2 \right </math>, <math>t \in [0, 1]</math>.</p> <p>...compute:  <math>g(t) = (1 - 3t + 3t^2) + k(1 - 2t + 2t^2) = (1+k) + (-3-2k)t + (3+2k)t^2 = (3+2k)t(t-1) + (1+k)</math>.</p> <p>Because <math>g(t) \geq 0</math> for all <math>t \in [0, 1]</math> with maximum at <math>(t = \frac{1}{2})</math>:  <math>g(\frac{1}{2}) = \frac{1}{4}(3+2k) + (1+k) = \frac{1}{4}(7+2k)</math>.</p> <p>Thus the minimum possible value of <math>g(t)</math> on <math>[0, 1]</math> is <math>g(\frac{1}{2})</math>.  <math>g(\frac{1}{2}) \geq 0 \iff \frac{1}{4}(7+2k) \geq 0 \iff k \geq -\frac{7}{2}</math>.</p> <p>Final Answer:  <math>\boxed{k \geq -\frac{7}{2}}</math>.</p> <p>These are exactly the real values of <math>k</math> for which there exists a real number <math>c</math> such that <math>f(c) = 0</math>.</p>	<p>Okay, we are to find all real numbers <math>k</math> such that there exists some real number <math>c</math> for which <math>f(c) = 0</math>, let's compute the resulting conditions ...</p> <pre> t**2 + (1-t)**2 + (1-t)**3 + k * (t**2 + (1-t)**2) f_simplified = sp.simplify(f) f_expanded = sp.expand(f_simplified) print("Expanded polynomial:", f_expanded) df = sp.diff(f_expanded, t) critical = sp.solve(sp.simplify(df), t) print("Critical points:", critical) - t_min = sp.solve(sp.diff(f, t), t) t_min = [sol for sol in t_min if sol.is_real] print(t_min) </pre> <p>Textual feedback:  Expanded polynomial: <math>2k t^2 + 2k t + 3t^2 - 3t + 1</math>  Critical points: <math>\{1/2\}</math></p> <p>From the code returned,  <math>f'(t) = 2(2k+3)t - (2k+3) = 0</math> for all <math>k \neq -\frac{3}{2}</math> (the case that eliminates the quadratic term). So the only interior critical point is <math>t = \frac{1}{2}</math>. Real roots exist exactly when <math>-(2k+1)(2k+3) \geq 0</math>, which gives the interval: <math>k \in [-\frac{3}{2}, -\frac{1}{2}]</math>. Since <math>k \neq -\frac{3}{2}</math>, the final answer is: <math>\boxed{-\frac{3}{2} &lt; k \leq -\frac{1}{2}}</math>.</p>	<p>Let's denote <math>s = \sin^2 x</math>, so <math>\cos^2 x = 1 - s</math>, and <math>s \in [0, 1]</math>. Then: <math>f(x) = \sin^6 x + \cos^6 x + k(\sin^4 x + \cos^4 x)</math> can be seen as: <math>f(s) = 1 - 3s + 3s^2 + k(2s^2 - 2s + 1) = (3 + 2k)s^2 + (-3 - 2k)s + (1 + k)</math>. We want to find all real <math>k</math> such that there exists <math>s \in [0, 1]</math> for which <math>f(s) = 0</math>. We can draw the function of this quadratic for different values of <math>k</math>, and consider under what conditions it has a zero point (intersects the x-axis) in the interval <math>[0, 1]</math>.</p>  <pre> - # Get unique k values where root exists unique_k = np.unique(results) - # Plot for a few k values to visualize plt.figure(figsize=(10, 6)) k_test = [-1.5, -1.0, -0.75, -0.5, 0.0] for k in k_test:     s_vals = np.linspace(0, 1, 1000)     f_vals = f(s_vals, k)     plt.plot(s_vals, f_vals, label=f'k={k}') - plt.title('Plot of f(s) for different k') plt.grid(True) plt.show() </pre> <p>From the plots we can see:  - For <math>k = -1</math>: <math>f(0) = 0</math> and <math>f(1) = 0</math>, so the function touches zero at the endpoints.  - For <math>k = -0.5</math>: <math>f(0.5) = 0</math>, so it touches zero at the vertex.  - For <math>k = -1.5</math>: <math>f(s) = -0.5</math> for all <math>s</math>, so it never touches zero.</p> <p>This verifies that the boundary values <math>k = 1</math> and <math>k = 0.5</math> are included, and the function is zero at some point in <math>[0, 1]</math> for all <math>k \in [-1, -0.5]</math>. Therefore, the set of all real numbers <math>k</math> for which there exists a real number <math>c</math> such that <math>f(c) = 0</math> is the interval <math>[-1, -1/2]</math>. We can now box the answer: <math>\boxed{[-1, -\frac{1}{2}]}</math>.</p>

Figure 4: Case study of reasoning behaviors. FIGR demonstrates more effective integration of visual feedback into the reasoning process, resulting in clearer intermediate reasoning and improved final answers, while the baseline models rely primarily on textual reasoning.

explicit control signal that regulates when visual reasoning should be invoked. (2) Removing visual feedback. This setting corresponds to text-only RL with code execution but without rendered image feedback. The model still performs multi-turn rollouts and executes code, but receives only textual feedback and its corresponding calling reward. This setting isolates the contribution of visual feedback in forming a stable reasoning loop. Both ablations lead to performance degradation compared to the full FIGR, demonstrating the effectiveness of the proposed components.

**Evolution of Active Visual Construction During Training.** We further analyze the emergence of visual-thinking behaviors on the BeyondAIME dataset, using four complementary metrics: response length (measured in tokens with the Qwen3-VL-32B-Instruct tokenizer), code ratio (fraction of samples with at least one code execution), average code lines (non-empty lines per code block), and code pass rate (fraction of code blocks that execute successfully).

(1) Removing the adaptive reward mechanism (w/o ARM) leads to a brief increase in early code usage. However, this behavior rapidly collapses: both code ratio and code length drop to zero as the model learns that indiscriminate code invocation does not improve outcomes. This pattern reflects an overuse–abandonment cycle in the absence of

effective reward guidance.

(2) In contrast, FIGR maintains a consistently high code ratio throughout training, accompanied by stable and substantially longer code blocks with a high execution pass rate. Executable visual feedback provides an external state that enables explicit validation of global structure, which cannot be reliably enforced through implicit textual representations alone. Consequently, visual thinking becomes a more reliable mechanism for validating the whole reasoning process, rather than an opportunistic or exploratory behavior.

#### 4.4 Case Study

Figure 4 compares Qwen3-VL-32B-Instruct, the variant without visual feedback, and FIGR on a representative example. The baseline model relies on textual reasoning and symbolic manipulation, requiring careful analytical handling of boundary cases, making it prone to subtle oversights. The variant without visual feedback introduces multi-turn rollouts and intermediate code execution, but receives only textual feedback. While this allows the model to verify intermediate numerical values or symbolic expressions, it provides no access to visual cues that reveal global structural properties of the problem (e.g., the overall shape of a function or the location of its zeros). Consequently, reasoning remains largely local and algebraic, and incorrect assumptions are difficult to detect or correct.

In contrast, **FIGR** closes the reasoning loop by incorporating explicit visual state, enabling it to observe global structural patterns and adjust subsequent reasoning steps accordingly. This leads to a more robust and interpretable solution. Overall, this case study shows that the advantage of **FIGR** does not stem merely from executing intermediate programs, but from actively integrating visual construction into the reasoning process. Compared with purely textual reasoning or execution with text-only feedback, active visual thinking provides complementary information that substantially improves reasoning reliability.

## 5 Conclusion

We introduce **FIGR**, which integrates executable visual construction into multi-turn reasoning via end-to-end reinforcement learning. By embedding controllable and revisable visual states within the reasoning loop, **FIGR** supports more reliable reasoning over global structural constraints that are difficult to maintain through text alone. Extensive experiments demonstrate that learning when and how to externalize intermediate structure as executable visual states improves reasoning stability and accuracy on complex mathematical problems.

## Limitations

While our results demonstrate the effectiveness of executable visual construction for complex reasoning, several limitations remain, suggesting directions for future work. The proposed method incurs additional computational cost due to multi-turn reasoning and code execution. This overhead is inherent in approaches that externalize intermediate states, motivating future efforts toward more efficient reasoning–execution loops through compact visual representations or adaptive control strategies. Executable visual construction is particularly beneficial for problems that require global structural or relational constraints. In contrast, tasks that can be easily solved through local or purely symbolic reasoning may benefit less from visual externalization. More precise identification of such task regimes remains an open question. Our evaluation focuses on mathematical reasoning benchmarks that emphasize structural consistency and verifiable outcomes. Extending this paradigm to other domains, including scientific reasoning, program synthesis, and long-term planning, represents a natural and promising direction for future work.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. [Gpt-4 technical report](#). *ArXiv preprint*, abs/2303.08774.
- AIME. 2024. American invitational mathematics examination (aime).
- AIME. 2025. American invitational mathematics examination (aime).
- AMC. 2023. American mathematics competitions (amc).
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, and 1 others. 2025a. [Qwen3-vl technical report](#). *ArXiv preprint*, abs/2511.21631.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025b. [Qwen2.5-vl technical report](#). *ArXiv preprint*, abs/2502.13923.
- Mark Chen. 2021. [Evaluating large language models trained on code](#). *ArXiv preprint*, abs/2107.03374.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. [Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks](#). *ArXiv preprint*, abs/2211.12588.
- Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, and 1 others. 2025. [Emerging properties in unified multimodal pretraining](#). *ArXiv preprint*, abs/2505.14683.
- Merlin Donald. 1993. *Origins of the modern mind: Three stages in the evolution of culture and cognition*. Harvard university press.
- Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. [Retool: Reinforcement learning for strategic tool use in llms](#). *ArXiv preprint*, abs/2504.11536.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *ArXiv preprint*, abs/2501.12948.
- Tanmay Gupta and Aniruddha Kembhavi. 2023. [Visual programming: Compositional visual reasoning without training](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 14953–14962. IEEE.

- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.
- Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, and 1 others. 2025. [Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning](#). *ArXiv preprint*, abs/2504.11456.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). *ArXiv preprint*, abs/2103.03874.
- 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](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. [Solving quantitative reasoning problems with language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Ang Li, Charles Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, and 1 others. 2025a. [Zebra-cot: A dataset for interleaved vision language reasoning](#). *ArXiv preprint*, abs/2507.16746.
- Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. 2024. [Chain of code: Reasoning with a language model-augmented code emulator](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. 2025b. [Dyfo: A training-free dynamic focus visual search for enhancing llms in fine-grained visual understanding](#). In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9098–9108.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. [Deepseek-v3 technical report](#). *ArXiv preprint*, abs/2412.19437.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Jean M Mandler. 2010. The spatial foundations of the conceptual system.
- Don Norman. 2024. *Things that make us smart*. Diverision Books.
- ByteDance Seed, Jiase Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, and 1 others. 2025. [Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning](#). *ArXiv preprint*, abs/2504.13914.
- Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. [Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *ArXiv preprint*, abs/2402.03300.
- Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruo Chen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. 2025. [Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6613–6629.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. [Hybridflow: A flexible and efficient rlhf framework](#). *arXiv preprint arXiv: 2409.19256*.
- Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. [Vipergpt: Visual inference via python execution for reasoning](#). In *IEEE/CVF International Conference*

- on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, pages 11854–11864. IEEE.
- Chameleon Team. 2024. [Chameleon: Mixed-modal early-fusion foundation models](#). *ArXiv preprint*, abs/2405.09818.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. [Llama: Open and efficient foundation language models](#). *ArXiv preprint*, abs/2302.13971.
- Barbara Tversky, Julie Heiser, Paul Lee, and Jeffrey M Zacks. 2019. Diagrams to augment cognition. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*, pages 57–57. Routledge.
- Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhui Chen. 2025. [Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning](#). *ArXiv preprint*, abs/2505.15966.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, and 1 others. 2025a. [Qwen-image technical report](#). *ArXiv preprint*, abs/2508.02324.
- Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and 1 others. 2025b. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977.
- Penghao Wu and Saining Xie. 2024. [V\\*: Guided visual search as a core mechanism in multimodal llms](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13084–13094. IEEE.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. [Qwen3 technical report](#). *ArXiv preprint*, abs/2505.09388.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and 1 others. 2025. [Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl](#). *ArXiv preprint*, abs/2505.15436.
- Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. [Deepeyes: Incentivizing "thinking with images" via reinforcement learning](#). *ArXiv preprint*, abs/2505.14362.

Model	AIME 2024	AIME 2025	Beyond AIME	MATH 500	AMC	Minerva Math	Olymp Bench	Avg.
<i>Large Language Models</i>								
Qwen3-235B-A22B (Thinking)	83.80	80.78	52.00	95.40	93.98	46.69	74.04	75.24
Qwen3-32B (Non-Thinking)	31.00	20.20	16.00	88.60	61.45	39.34	52.08	44.10
Qwen3-32B (Thinking)	81.40	72.90	40.00	97.30	93.98	45.22	71.96	71.82
<i>Large Vision-Language Models</i>								
GLM-4.5V (108B)	76.20	67.34	47.00	96.20	87.95	38.60	70.03	69.62
Qwen3-VL-8B-Instruct	61.46	46.20	30.00	94.20	81.93	40.07	67.21	60.15
Qwen3-VL-32B-Instruct	73.33	66.20	43.00	93.60	84.34	41.18	69.73	67.34
<i>Unified Multimodal Models</i>								
Bagel-7B-MoT	10.00	3.33	1.00	56.00	22.89	9.56	26.11	18.70
Bagel-Zebra-CoT (7B)	0.00	5.55	0.00	20.60	9.64	2.94	8.46	6.74
<i>Tool-Augmented Vision-Language Models</i>								
DeepEyes	6.98	2.34	0.00	11.00	18.07	5.15	13.65	8.17
Chain-of-Focus	1.30	0.00	0.00	1.20	0.00	0.74	0.45	0.53
Text-only RL	73.33	69.22	46.00	94.40	87.95	43.38	70.33	69.23
<b>FIGR (ours)</b>	<b>79.58</b>	<b>79.32</b>	<b>54.00</b>	<b>95.00</b>	<b>93.98</b>	<b>44.49</b>	<b>72.40</b>	<b>74.11</b>
<b>Gains</b>	<b>+6.25</b>	<b>+13.12</b>	<b>+11.00</b>	<b>+1.40</b>	<b>+9.64</b>	<b>+3.31</b>	<b>+2.67</b>	<b>+6.77</b>

Table 3: Full experimental results (%) on seven mathematical reasoning benchmarks compared with several competitive baselines. **Green-colored** font indicates improvement over the baseline Qwen3-VL-32B-Instruct.

## A Dataset Details

We evaluate models on a suite of challenging mathematical reasoning benchmarks, spanning competition problems, standardized mathematics datasets, and advanced reasoning tasks:

(1) **AIME 2024** (AIME, 2024) dataset contains 30 problems from the American Invitational Mathematics Examination 2024. Each problem requires multi-step reasoning across algebra, number theory, combinatorics, geometry, and probability.

(2) **AIME 2025** (AIME, 2025). Like the AIME 2024 dataset, the AIME 2025 dataset comprises the 30 problems from the 2025 AIME competitions.

(3) **Beyond AIME** (Seed et al., 2025) is a recently proposed benchmark of 100 problems. It is designed to push beyond standard AIME problems by emphasizing questions that reduce memorization and increase the complexity of reasoning.

(4) **MATH 500** (Hendrycks et al., 2021; Lightman et al., 2024). The MATH dataset contains 12,500 problems with step-by-step solutions spanning diverse mathematical topics. For evaluation focused on high-difficulty items, the MATH 500 subset (500 problems) is often used.

(5) **AMC** (AMC, 2023). The American Mathematics Competitions (AMC) dataset consists of 83 problems from the AMC series, a set of standardized mathematics contests administered annually.

(6) **MinervaMath** (Lewkowycz et al., 2022) includes 272 undergraduate-level quantitative reason-

ing problems that require logical deductions and multi-step solutions, serving as a more advanced testbed beyond high-school competition problems.

(7) **OlympiadBench** (He et al., 2024) is an Olympiad-level bilingual multimodal scientific benchmark that includes thousands of challenging mathematics and physics problems compiled from international competitions, with detailed annotations for step-by-step reasoning. In this work, we specifically evaluate the math-focused English comprehensive subsets from OlympiadBench: *OE\_TO\_maths\_en\_COMP*. This subset is an English open-ended, text-only, comprehensive math subset containing 674 problems, emphasizing rigorous mathematical reasoning in text form.

## B Prompt Templates

### B.1 Prompt Template for Suitability Classifier

We provide a dedicated prompt template for the suitability classifier as introduced in Section 3.3 and Section 4.1, which is used to determine whether a given problem is suitable for diagrammatic reasoning. As shown in Figure 5, the template instructs the model to assess the necessity and usefulness of visual sketches based solely on the problem statement. For each training instance, we run the classifier independently eight times and use majority voting to assign the suitability label.

---

**Prompt Template for Suitability Classifier**

---

**Instruction:**

You are a classifier. Your task is to decide whether solving a given problem would significantly benefit from drawing a diagram/sketch (i.e., “a picture is worth a thousand words,” “think while drawing”).

Labeling rule:

Output 1 if a diagram is very helpful or almost necessary for understanding/solving the problem (e.g., geometry, spatial relations, motion paths, graphs of functions, logical/structural relations that are hard to follow in text alone).

Output 0 if a diagram adds little or no value (e.g., simple algebraic manipulation, direct factual recall, definition/knowledge questions, straightforward calculations).

Only output 1 or 0, nothing else.

**Examples:**

Problem: Let the circles  $k_1$  and  $k_2$  intersect at two distinct points  $A$  and  $B$ , and let  $t$  be a common tangent of  $k_1$  and  $k_2$ , that touches  $k_1$  and  $k_2$  at  $M$  and  $N$ , respectively. If  $t \perp AM$  and  $\angle NMB = 2\angle A$ , evaluate  $\angle NMB$ .

Label: 1

Problem: Let  $a, b$  be two distinct real numbers and let  $c$  be a positive real number such that  $a^4 - 2019a = b^4 - 2019b = c$ . Prove that  $\sqrt{c} < a < b < 0$ .

Label: 0

**Problem:**

{problem}

**Assistant:**

---

Figure 5: Prompt template for the suitability classifier.

## B.2 Prompt Template for Reasoning

We adopt three different categories of prompt templates corresponding to the reasoning settings. (1) As shown in Figure 6, a multi-turn prompt is used for **FIGR**, the variant without visual feedback, and the variant without ARM. Models can iteratively generate output and receive feedback from the interpreter. (2) As shown in Figure 7, a standard single-turn prompt is adopted for the text-only RL baseline and other models, requiring the model to directly output the final answer without intermediate interactions. (3) For DeepEyes and Chain-of-Focus, we adopt the prompt templates recommended by their original papers.

## C Additional Experimental Results

As shown in Table 3, we present the full experimental results, including unified multimodal models and tool-augmented vision-language models that are partially reported in the main context.

Prompt Template for Multi-turn CoT
<p><b>Instruction:</b> You are a helpful assistant. You can call functions to assist with the user query. Important: You must call only one function at a time. After each function call, wait for the execution result before making the next function call if needed.</p> <p><b>Tools:</b> You are provided with function signatures within <code>&lt;tools&gt;&lt;/tools&gt;</code> XML tags:  <code>&lt;tools&gt;</code>  <code>{"type": "function", "function": {"name": "code_interpreter", "description": "A tool for executing code.", "parameters": {"type": "object", "properties": {"code": {"type": "string", "description": "The code to execute."}}, "required": ["code"]}}</code>  <code>&lt;/tools&gt;</code></p> <p>For each function call, return a json object with function name and arguments within <code>&lt;tool_call&gt;&lt;/tool_call&gt;</code> XML tags:  <code>&lt;tool_call&gt;</code>  <code>{"name": &lt;function-name&gt;, "arguments": &lt;args-json-object&gt;}</code>  <code>&lt;/tool_call&gt;</code></p> <p><b>Answer Format:</b> Think first, call <code>**code_interpreter**</code> if needed, then answer. Output the final answer in the following format: <code>\boxed{{The final answer goes here.}}</code> Now, let's solve the problem step by step:</p> <p><b>User Question:</b> {question}</p> <p><b>Assistant:</b></p>

Figure 6: Prompt template for multi-turn CoT.

Prompt Template for Single-turn CoT
<p><b>Instruction:</b> You are a helpful assistant. After you finish all reasoning, present the final result.</p> <p><b>Answer Format:</b> Let's solve the problem step by step and output the final answer in the following format: <code>\n\n\boxed{{The final answer goes here.}}</code></p> <p><b>User Question:</b> {question}</p> <p><b>Assistant:</b></p>

Figure 7: Prompt template for single-turn CoT.