Play to Generalize: Learning to Reason Through Game Play

Yunfei Xie[1], Yinsong Ma[2], Shiyi Lan[3], Alan Yuille[2], Junfei Xiao[2*], Chen Wei[1†]

[1]Rice University, [2]Johns Hopkins University, [3]NVIDIA

Website https://yunfeixie233.github.io/ViGaL

Code & Model & Data https://github.com/yunfeixie233/ViGaL

Abstract

Developing generalizable reasoning capabilities in multimodal large language models (MLLMs) remains challenging. Motivated by cognitive science literature suggesting that gameplay promotes transferable cognitive skills, we propose a novel post-training paradigm, Visual Game Learning or ViGaL, where MLLMs develop out-of-domain generalization of multimodal reasoning through playing arcade-like games. Specifically, we show that after training a 7B-parameter MLLM via reinforcement learning (RL) on simple arcade-like games, *e.g.*. Snake, significantly enhances its downstream performance on multimodal math benchmarks like MathVista, and on multi-discipline questions like MMMU, *without seeing any worked solutions, equations, or diagrams during RL*, suggesting the capture of transferable reasoning skills. Remarkably, our model outperforms specialist models tuned on multimodal reasoning data in multimodal reasoning benchmarks, while preserving the base model's performance on general visual benchmarks, a challenge where specialist models often fall short. Our findings suggest a new post-training paradigm: synthetic, rule-based games can serve as controllable and scalable pre-text tasks that unlock generalizable multimodal reasoning abilities in MLLMs.

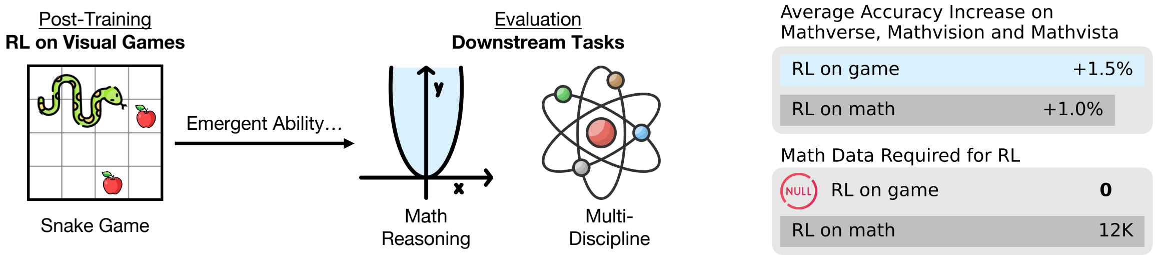


Figure 1 | **Overview of ViGaL.** *Left*: We propose a novel post-training paradigm where MLLMs are tuned via RL to play arcade-style games such as Snake [32]. We demonstrate that gameplay post-training enables MLLMs to achieve *out-of-domain* generalization, enhancing their performance on downstream multimodal reasoning tasks requiring math, spatial and multi-discipline reasoning, without using in-domain math or multi-displine data during RL. *Right*: Our ViGaL (RL on game) achieves higher average accuracy increase than MM-Eureka [48] (RL on math) across three multimodal math benchmarks. This is notable because MM-Eureka uses RL on large-scale, curated math datasets, while ViGaL only uses game data. Details are in Tab 2.

*Project Lead; †Corresponding Author

1. Introduction

Games, beyond their entertainment value, provide rich and diverse structured environments for developing and studying general reasoning and problem-solving abilities. Humans from early childhood acquire foundational cognitive skills through diverse game-like activities such as arranging objects, navigating spaces, and manipulating tools. These experiences foster essential building blocks of abstract thinking, including pattern recognition, spatial reasoning, and causal inference [8, 9]. In cognitive science, games are used as experimental platforms to reveal the inductive biases of the human mind [2, 3], such as planning depth in the game Four-in-a-Row [63], or the cognitive basis of tool use through the game Virtual Tools [4].

AI agents, too, benefited from games resembling aspects of human play. These environments encourage exploration, robustness to sparse rewards, and learning from multimodal inputs. For example, emergent tool use has been observed in agents trained via hide-and-seek [7], and Atari gameplay has been incorporated into training generalist agents [52]. By learning in these environments, AI systems develop robust and transferable reasoning capabilities.

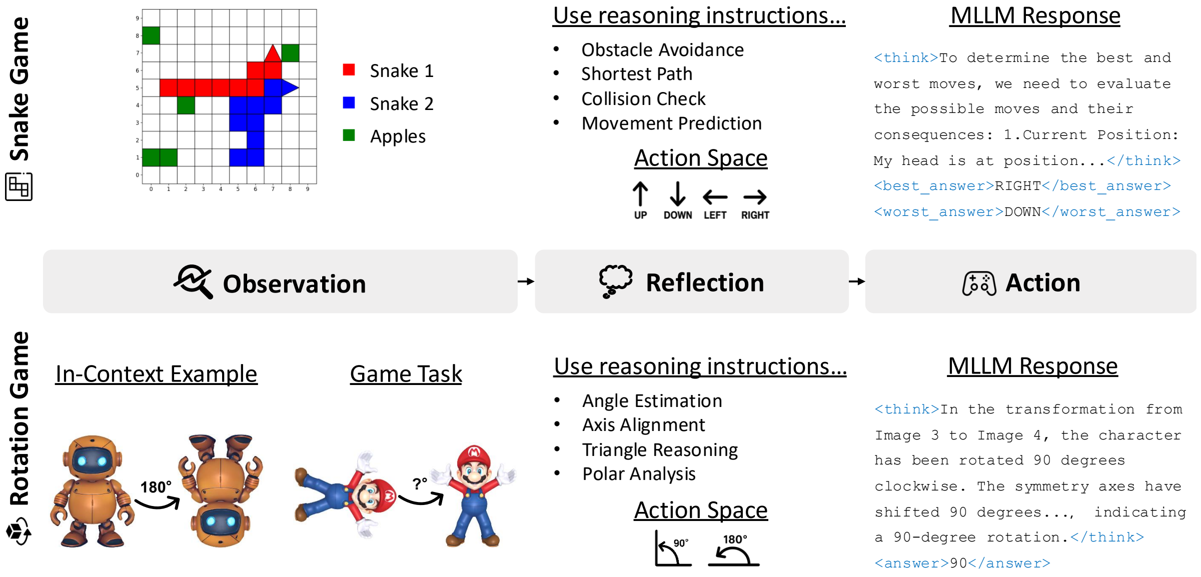
In this work, we specifically study the use of gameplay in the context of post-training multimodal large language models to effectively reason. Recent work has shown that post-training with Reinforcement Learning (RL) can unlock reasoning behaviors from their base models [16, 49]. These RL-trained models are able to successfully "think before they speak", generating internal chain-of-thought traces before outputting a final answer.

More importantly, growing evidence suggests that RL often generalizes more robustly to out-of-distribution samples than supervised fine-tuning (SFT), another widely used post-training approach. For example, models trained with RL on CLEVR [31] generalize to more challenging Super-CLEVR benchmark [40], models trained on math problems extend reasoning to physics questions [48], and agents trained to navigate one environment successfully adapt to novel locations [14]. In each case, RL-trained models consistently outperform their SFT counterparts.

While these results demonstrate promise of *out-of-distribution* generalization, they typically remain within a single domain. The source and target tasks still belong to the same family, such as STEM questions [48] or spatial navigation [14]. In this work, we explore the potential of a stronger form of *out-of-domain* generalization: transferring from one domain to an entirely different one, specifically, from gameplay to math questions.

As illustrated in Fig. 1, we show that post-training a 7B-parameter multimodal model, Qwen2.5-VL-7B [6], to play simple arcade-style games like Snake [32] (1) generalizes to nail *out-of-distribution* unseen Atari games (Sec. 2.3), and (2) obtains enhanced *out-of-domain* capabilities on multimodal math benchmarks, *e.g.*, MathVista [44], and multi-discipline question answering, *e.g.*, MMMU [71]. Despite never seeing any worked solutions, equations, or diagrams during RL, our model outperforms not only large-scale industrial systems like GPT-4o [30], but also specialist models post-trained on in-domain datasets for reasoning (Tabs. 2 and 3). Moreover, our model obtains improvements on multimodal reasoning benchmarks without sacrificing its general visual capabilities, a challenge for domain specialist models (Tab. 4). Interestingly, recent works challenge the necessity of ground-truth labels of in-domain questions for RL [54, 75], while our approach suggests that in-domain questions themselves may not be required.

Why does it work? We hypothesize that gameplay encourages generalizable cognitive primitives or skills that are transferable to multimodal reasoning benchmarks such as spatial understanding and sequential planning. Unlike SFT or RL on math questions, which could reinforce memorization on training data [14, 73], gameplay training may incentivize more flexible representations and strategies. Supporting this view, our ablation studies reveal that both the prompt and reward designs play critical roles in enabling effective learning (Sec. 3.2).



```
<think>To determine the best and
worst moves, we need to evaluate
the possible moves and their
consequences: 1.Current Position:
My head is at position...</think>
<best_answer>RIGHT</best_answer>
<worst_answer>DOWN</worst_answer>
```

```
<think>In the transformation from
Image 3 to Image 4, the character
has been rotated 90 degrees
clockwise. The symmetry axes have
shifted 90 degrees...,  indicating
a 90-degree rotation.</think>
<answer>90</answer>
```

Figure 2 | **Post-training MLLMs to reason through RL with games.** We propose post-training MLLMs via RL by playing visual games. We demonstrate this with two games: the classic arcade game Snake [32], and Rotation, a self-designed task to investigate spatial reasoning. In each game, the model receives multimodal inputs and follows reasoning instructions, *e.g.*, path planning in Snake, angle estimation in Rotation. It reflects to choose an action, outputs its chain-of-thoughts and decision, *e.g.*, best/worst move or predicted angle, and receives a reward. Through game playing, the model obtains reasoning abilities that transfer to downstream multimodal reasoning tasks such as math and multi-discipline question answering (Fig. 3).

We also find that different games emphasize distinct reasoning skills: Snake, a 2D grid game where the player maneuvers the “snake” to avoid collisions and reach apples, promotes performance on multimodal math questions concerning 2D coordinates, while Rotation, a puzzle to identify the rotating angle of 3D objects, performs better on angle and length related ones (Fig. 5). Furthermore, training on both tasks together leads to consistently better performance on downstream multimodal reasoning benchmarks than training on either game alone, suggesting the scalable possibility of games (Tab. 2).

These results suggest a new post-training paradigm. Beyond collecting domain-specific data, we can also design scalable and controllable pre-text games that unlock desired reasoning behaviors transferable to downstream tasks. Synthetic game environments provide structured, rule-based reward signals with high controllability, enabling stable RL learning through difficulty scheduling. There are contemporary works studying the properties of reasoning models with game environments, taking advantage of its controllability [56], while we emphasize on its out-of-domain generalization capability. Scaling the data in these environments is also significantly easier than collecting human-annotated data. Altogether, these findings indicate a promising paradigm of post-training with synthetic tasks such as games, reminiscent of the rise of self-supervised learning in vision and language [18, 27, 51], where pretraining on synthetic yet principled pre-text tasks leads to broad generalization.

The following sections are organized as follows: In Sec. 2, we focus on game tasks, introducing how to post-train with RL on games and show improvements on unseen games to demonstrate out-of-distribution generalization. In Sec. 3, we focus on out-of-domain generalization evaluation, further showing training on visual games brings improvement to out-of-domain generalization on unseen visual reasoning tasks. In Sec. 4, we summarize recent developments of reinforcement learning and generalization in MLLMs and highlight how our ViGaL differs by leveraging simple games to achieve stronger generalization.

2. Reinforcement Learning on Visual Games

In this section, we introduce ViGaL, a novel post-training paradigm designed to enhance generalization capabilities. Sec. 2.1 describes the Snake and Rotation game environments used for training and evaluation. Sec. 2.2 outlines the reinforcement learning algorithm employed in our framework. Sec. 2.3 presents implementation details and provides a comprehensive evaluation on both in-distribution and out-of-distribution games.

2.1. Game Environment

As show in Fig. 2, under our ViGaL paradigm, the model is trained in a game environment where it receives states from game environment, outputs next actions, and obtains rewards as feedback from the environment. Formally, each task, given an instruction I, can be formulated as a partially observable Markov decision process (POMDP): $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, \Omega)$, where $\mathcal{S}$ is the set of possible environment states, $\mathcal{O}$ is the set of observations available to the model, and $\mathcal{A}$ represents actions model can do in this game environment. $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state transition function, while R is a binary reward from the environment representing the correctness of action. Due to partial observability, the agent perceives only observations $o = \Omega(s)$.

We design two distinct games, Snake and Rotation, to study the proposed paradigm (Fig. 2), each targeting different MLLM capabilities. The Snake game is inspired by how competition can incentivize reasoning abilities in MLLMs [20]. It focuses on strategic decision-making by challenging the model to choose appropriate actions while competing with another snake. Meanwhile, the Rotation game draws inspiration from rotation-angle prediction as a supervised pre-text task in self-supervised learning [26]. This game evaluates the MLLM's visual perception capabilities, particularly in understanding complex 3D spatial transformations. Through these complementary games, we can systematically explore and improve reasoning and perception, two distinct and fundamental aspects of MLLM abilities.

Snake. We set up a dual-snake game based on SnakeBench [32]. Each model controls one snake independently. The objective of each snake is to reach apples and score points and outcompete the other. At time t, the environment state s^t contains the coordinates of the snakes $(x^t_{s_i}, y^t_{s_i})$ for snake $i \in \{1, 2\}$, the coordinates of the apple (x^t_a, y^t_a), and the last moves A^{t-1}_i chosen by each snake. These elements are positioned on a 10×10 game board. In each round t, each snake selects its next move A^t_i from $\{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$. A snake dies if it collides with itself, the other snake, or the boundary of the board. If one snake dies, the other snake wins. If both snakes die simultaneously, the snake with the higher score wins. Unlike SnakeBench [32] which uses only text to represent the game state, we use both an image of the game board and text descriptions as observation $o^t = \Omega(s^t)$ for enhanced representation.

Rotation game. We design a rotation game to study the spatial reasoning capabilities of MLLMs. We present the model with two views of the same 3D object: an initial view I_{init} and a rotated view I_{rot}. The rotated view is created by rotating the 3D object from its initial orientation by either 90° or 180° around the z-axis, which points toward the viewer. The task is to determine which degree, 90° or 180°, is applied to transform the object from the initial orientation to the rotated orientation. To guide the model's reasoning, we provide an in-context example consisting of another image pair with a known rotation angle. Similar to Snake game, we provide observations with both images and text.

2.2. Rule-Based Reinforcement Learning

We apply rule-based RL to directly post-train MLLMs for visual games, without relying on supervised learning as a warm up. The algorithm is described as follows:

Reward design. Instead of relying on outcome- or process-based reward models, following previous approaches [29, 76], we use a simple rule-based reward function to avoid reward hacking [25] and help the model learn how to play the games effectively.

This reward function has two components: an accuracy reward and a format reward. The total reward r is computed as the sum of an accuracy reward and a format reward $r = r_{\text{accuracy}} + r_{\text{format}}$. The accuracy reward r_{accuracy} is 1 if the answer is correct, and 0 otherwise.

The format reward r_{format} checks whether the response follows a task-specific format: $r_{\text{format}} = 0.1$ if the response is correctly formatted, and $r_{\text{format}} = 0$ otherwise. For Snake game, the desired format is:

`<think>...</think><best_answer>...</best_answer><worst_answer>...</worst_answer>.`

As suggested by the format, we encourage the model to predict both a positive move that moves toward the apple and a negative move that leads to failure. This reward encourages contrastive decision-making, which not only improves the model's gameplay abilities but also boosts downstream reasoning performance on visual math benchmarks. We ablate the effect in Tab. 5b. For the rotation task, the required format is simply `<think>...</think><answer>...</answer>`.

Advantage estimation and policy update. We employ REINFORCE Leave-One-Out (RLOO) algorithm [1, 35] in our RL training phase. We do not incorporate KL divergence regularization in our implementation, following the technique proposed in Group Policy Gradient [15]. Without KL constraints that limit how much the policy can change, the model may explore the solution space more freely, potentially discovering better reasoning strategies. This design choice allows our model to adapt more flexibly during our RL phase.

Text prompt design. While the model takes images as input to understand the current state of the game, we design a structural text prompt framework to also provide game guidance. Our game prompts consist of two parts: (1) game settings and (2) reasoning instructions. (1) To help the model understand the game environment, we describe the background, current game state, rules, goals, action space, *etc.* in text besides the input image. (2) In the reasoning instruction part, we provide specific thinking guidance since games can be approached with various thinking chains. To encourage broader thinking, we implement different types of reasoning instructions to guide decision-making process. Specifically, we used GPT-4o [30] to synthesize mathematical thinking instructions for Snake, such as "`finding the nearest apple by calculating Manhattan distances`", and spatial thinking instructions for Rotation, for example, "`identify major symmetry axes in the original image`". As shown in Fig. 4a, these reasoning instructions help models to lengthen responses or chains of internal thinking traces. With reasoning instructions for games, the obtained reasoning abilities generalize to downstream evaluation on visual math questions (Tab. 5a). The details of text prompt design, including the reasoning instructions used, are in Appendix Sec. A.2.

Controlling game difficulty. Thanks to using the synthetic game data engine, we can flexibly generate large-scale training data with precisely controlled difficulty levels. This completely eliminates the need for extensive data filtering strategies used in previous rule-based RL work training on domain-specific data like math [5, 48], where difficulty is hard to define and filtering can significantly reduce dataset size. In Snake, we define difficulty based on snake length, where longer snakes create more complex game situations and more constrained movement options, closely aligning with how humans perceive difficulty when playing Snake ourselves. In Rotation, difficulty is determined by the rotation angle between two images, where smaller angle differences present greater perceptual challenges. Based on empirical results, we established optimal difficulty parameters for RL training, which we ablate in Tab. 5c. This controlled progression of difficulty enables more effective learning trajectories.

Model	Wins (/10)
ViGaL vs.	
Qwen2.5-VL-7B	9
Qwen2.5-VL-72B	7
Llama-4-Maverick	7
Gemini-2.5-Pro	8
Claude-3.7-Sonnet	6
GPT-4o	8
o4-mini	6

(a) Snake game.

Model	Acc. (%)
ViGaL	71.9
Qwen2.5-VL-7B	47.4
Qwen2.5-VL-72B	52.1
Llama-4-Maverick	66.2
Gemini-2.5-Pro	51.0
Claude-3.7-Sonnet	65.6
GPT-4o	61.5
o4-mini	70.8

(b) Rotation game.

Game	ViGaL	Qwen2.5-VL-7B
Space Invaders	280.0	85.0
Ms. Pacman	1370.0	670.0
Seaquest	80.0	60.0
Alien	540.0	450.0
Frogger	7.0	5.0
Breakout	0.0	9.0
Pong	-26.0	-26.0
Cumulative Reward	**2251.0**	**1253.0**

(c) Atari game.

Table 1 | **Game Performance.** (a) In Snake, ViGaL consistently achieves the highest win rate(6-9 wins out of 10 matches), further surpassing larger proprietary models. (b) In Rotation, ViGaL demonstrates overall best performance with the best accuracy compared to leading commercial language models. (c) In the Atari Games, ViGaL training on Snake and Rotation games shows remarkably impressive zero-shot generalization to unseen Atari games, achieving nearly *double* the cumulative reward compared to Qwen2.5-VL-7B.

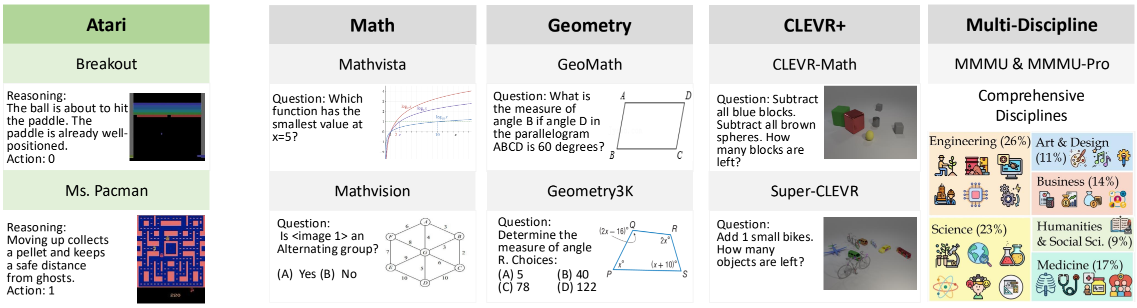
2.3. Implementation and Evaluation on Games

Implementation details. We employ Qwen2.5-VL-7B-Instruct [6] as our base model. We follow DeepSeek-R1 [16], using a combination of rule-based format rewards and accuracy rewards, with RLOO [1, 35] as the core RL algorithm. We implement our training within a multimodal input RL framework based on OpenRLHF [28]. For hyperparameters, we adopt the default settings from MM-Eureka [48], including a global batch size of 128, a rollout batch size of 128, a rollout temperature of 1.0, and a learning rate of $1e^{-6}$. Training uses 6 A100-80G GPUs.

Game training data. We build custom game environments to collect training data for our experiments. For Snake game, we leverage SnakeBench [32] as our data engine. This environment allows us to input actions to control snake movements and generate gameplay trajectories. To create meaningful gameplay data, we implement a policy network based on Proximal Policy Optimization (PPO) [53] with a linear output layer. This network continuously generates actions for two snakes that attempt to collect apples while avoiding death, enabling automatic capture of diverse gameplay trajectories for RL training. For the Rotation game, we utilize Hunyuan3D [59], a model that generates 3D meshes based on images or text instructions. We render each mesh into 2D images from different orientations, creating image pairs with associated rotation angles as ground truth labels for RL training data.

Our comprehensive data generation pipeline enables producing training samples at any desired scale with fully customized settings. For our experiments, we synthesize 36K samples for Snake and 36K samples for Rotation, which have shown to be sufficient for convergence. Further details of data synthesis are in Appendix Sec. A.1.

Competing with leading models on Snake and Rotation. To evaluate the game capabilities of ViGaL models, we initialize these environments in diverse states that were not seen during training. For Snake in Tab. 1a, we randomly initialize the game 10 times and have two models compete against each other to directly measure the win count of each model. For Rotation in Tab. 1b, we measure the rotation angle prediction accuracy on a comprehensive validation set consisting of 3D object meshes unseen during training. Our 7B-parameter model consistently outperforms proprietary models in both Snake and Rotation games. These results confirm that RL effectively unlocks the ability of a small 7B model to excel in visual games that require environmental understanding, reasoning, planning, and interactive decision-making.


Atari
Breakout
Reasoning:
The ball is about to hit the paddle. The paddle is already well-positioned.
Action: 0
Ms. Pacman
Reasoning:
Moving up collects a pellet and keeps a safe distance from ghosts.
Action: 1
Math
Mathvista
Question: Which function has the smallest value at x=5?
Mathvision
Question:
Is <image 1> an Alternating group?
(A) Yes (B) No
Geometry
GeoMath
Question: What is the measure of angle B if angle D in the parallelogram ABCD is 60 degrees?
A
D
B
C
Geometry3K
Question:
Determine the measure of angle R. Choices:
(A) 5 (B) 40
(C) 78 (D) 122
(2x − 16)°
Q
R
2x°
x°
(x + 10)°
P
S
CLEVR+
CLEVR-Math
Question: Subtract all blue blocks. Subtract all brown spheres. How many blocks are left?
Super-CLEVR
Question:
Add 1 small bikes. How many objects are left?
Multi-Discipline
MMMU & MMMU-Pro
Comprehensive Disciplines
Engineering (26%)
Art & Design (11%)
Business (14%)
Science (23%)
Humanities & Social Sci. (9%)
Medicine (17%)

(a) Out-of-distribution games.

(b) Out-of-domain tasks.

Figure 3 | **Samples from our generalization reasoning benchmarks.** We evaluate the proposed ViGaL with two types of generalization: (a) *out-of-distribution* generalization, where models trained on our visual games are tested on unseen Atari games [66]; and (b) *out-of-domain* generalization, where models trained only on game tasks are evaluated on diverse multimodal reasoning tasks including mathematical reasoning, geometric problem-solving, 3D understanding on CLEVR+ and multi-discipline reasoning on MMMU series.

Out-of-distribution generalization to Atari games. To evaluate out-of-distribution generalization, we test ViGaL on Atari-GPT [66], a benchmark for evaluating MLLMs as decision-making agents in Atari video games such as in Fig. 6. The benchmark consists of seven different Atari games, with detailed settings in Appendix Sec. B.1. We follow most settings and prompts from Atari-GPT, with a small modification providing explicit JSON output to ensure format correctness for all models. Following Atari-GPT [66], we report cumulative reward over 1K steps as the evaluation metric, where higher rewards indicate better performance. As shown in Tab. 1c, ViGaL demonstrates significant cumulative reward improvement on Atari games despite being trained only on Snake and Rotation games. This is particularly notable because Atari games differ substantially from our training games in both visual appearance and gameplay strategies. These results suggest that our rule-based RL training approach enables strong out-of-distribution generalization to entirely unseen game environments.

3. Visual Reasoning Generalization

Evaluation collection. To obtain a clearer picture of the various facets of MLLM performance, we follow prior studies [39, 61] and systematically and carefully divide existing benchmarks into two broad groups: (i) *reasoning-oriented benchmarks*, which require multi-step or mathematical reasoning to solve the problems, and (ii) *general-purpose perception benchmarks*, which primarily assess broad visual understanding and perception abilities.

For reasoning-oriented benchmarks, we comprehensively evaluate the visual reasoning generalization capabilities of RL through gaming on a diverse collection of tasks that specifically demand advanced visual reasoning skills, including math-focused tasks like Math and Geometry, and other comprehensive reasoning benchmarks beyond math, like CLEVR+ and Multi-Discipline. Fig. 3b illustrates specific examples from each benchmark.

- *Math* evaluates multimodal math reasoning with widely-used datasets: MathVista (testmini) [44], MathVerse (testmini) [74], and MathVision (test) [65]. MathVista offers diverse problems spanning VQA, logic, algebra, and geometry; MathVerse emphasizes algebraic and geometric image comprehension; MathVision tests abstract visual reasoning.
- *Geometry* evaluates structural interpretation skills across mathematical diagrams, medical images, charts, and architectural layouts. It uses datasets GeoMath (Geo170K [24],

Math360K [55]) and Geometry3K [45], featuring both choice and non-choice questions. Following Reason-RFT [57], we test with 820 GeoMath and 800 Geometry3K samples.

- *CLEVR+* evaluates the integration of mathematical and spatial reasoning skills through challenging arithmetic problems in complex 3D block-based scenes, including sub-tasks on CLEVR-Math [41] and Super-CLEVR [40]. Following Reason-RFT [57], we use 1K test samples from each of CLEVR-Math and Super-CLEVR.
- *Multi-Discipline* evaluates college-level expert knowledge across six disciplines: Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering. We follow the evaluation setting of MMMU [71] val set (900 questions) and MMMU-Pro [72] overall score (average of standard 10-option and vision-only settings).

For general-purpose perception benchmarks, we systematically evaluate comprehensive visual capabilities. Following previous work, these benchmarks are categorized into three distinct types: General, Vision-Centric, and OCR & chart. Specifically, for General, we evaluate MuirBench [64] for multi-image understanding and CRPE [33] for relation understanding. For Vision-Centric benchmarks, we assess MMVP [62], RealWorldQA [68], MMStar [12], MME [22], and BLINK [23] to thoroughly evaluate perception, real-world understanding, and multi-modal capabilities. For OCR & Chart understanding, we specifically use AI2D [34] for diagram understanding, SEED-Bench-2-Plus [37] for text-rich visual comprehension, DocVQA [47] for document understanding, and OCRBench [43] for comprehensive OCR evaluation.

3.1. Main Results

Zero-shot generalization from gameplay to multimodal reasoning. Our approach consistently shows remarkable generalization capabilities on mathematical and other reasoning tasks, despite having no direct exposure to in-domain training data during RL post-training. As shown in Tab. 2, our method notably outperforms models specifically RL-trained on mathematical tasks. For instance, ViGaL Snake + Rotation achieves 0.5% higher accuracy than MM-Eureka-Qwen-7B [48] on Math and 28.7% on Geometry, even though MM-Eureka-Qwen-7B was explicitly trained on high-quality mathematical and geometry datasets.

This strong generalization extends beyond mathematics. Tab. 3 shows that ViGaL Snake + Rotation outperforms R1-OneVision-7B [70] by 5.4% on average across MMMU series benchmarks, which test multi-disciplinary reasoning. This is particularly notable since R1-OneVision-7B was trained on a carefully curated comprehensive dataset spanning multiple subjects.

These empirical results suggest that gameplay-based post-training develops fundamental reasoning capabilities that transfer more effectively than direct RL training on diverse task-specific datasets. Moreover, the gameplay environment appears to encourage general problem-solving strategies that consistently generalize well to out-of-domain tasks.

Blending multiple games enhances generalization. As shown in Tab. 2, post-training on Snake achieves the best performance on the CLEVR+ benchmark, while training on Rotation yields stronger results on geometry reasoning. Their comparative strengths are further illustrated in Fig. 5. Notably, training the model on both Snake and Rotation games together enables it to learn complementary skills from each environment, improving the overall benchmark average to 63.1%. These findings suggest that combining diverse game environments can drive meaningful performance gains. This demonstrates the potential of Visual Gaming Learning as a promising training paradigm for enhancing generalizable reasoning, without requiring large-scale domain-specific data. Expanding the diversity of games during training consistently scales performance across a wide range of visual reasoning tasks.

Model		Math				Geometry		
	Avg.	Avg.	MathVista	MathVerse	MathVision	Avg.	GeoMath	Geo3K
Proprietary Model								
GPT-4o [30]	47.5	48.1	61.4	50.2	30.4	46.8	50.2	43.5
Gemini-2.0-Flash [58]	55.4	56.4	73.4	54.6	41.3	54.4	55.3	53.5
General Multimodal Language Model								
InternVL2.5-8B [13]	48.2	41.2	64.4	39.5	19.7	55.2	63.0	47.3
Llava-OV-7B [36]	–	–	63.2	26.2	–	60.7	77.6	43.7
Qwen2.5-VL-7B [6]	46.3	47.7	68.0	49.0	26.0	44.8	44.0	45.6
Multimodal Reasoning Model Post-Trained on Qwen2.5-VL-7B								
R1-Onevision-7B [70]	40.9	46.8	64.1	46.4	29.9	35.0	45.4	24.5
R1-VL-7B [11]	40.9	42.7	63.5	40.0	24.7	39.0	42.0	36.1
MM-Eureka-Qwen-7B [48]	39.3	50.1	73.0	50.3	26.9	28.4	53.1	3.8
Reason-RFT-Zero-7B [57]	46.5	38.1	60.7	35.3	18.3	54.9	55.0	54.8
VLAA-Thinker-7B [10]	51.3	48.7	68.0	51.7	26.4	53.9	51.1	56.6
OpenVLThinker-7B [17]	52.1	47.8	70.2	47.9	25.3	56.4	49.2	63.5
ViGaL Snake	51.6	49.4	70.7	51.1	26.5	55.0	49.9	60.0
ViGaL Rotation	52.8	49.3	71.2	50.4	26.3	**57.9**	51.7	64.1
ViGaL Snake + Rotation	**53.9**	**50.6**	71.9	52.4	27.5	57.1	51.0	63.3

Table 2 | **Main results on multimodal mathematical benchmarks.** We primarily compare with multimodal reasoning models post-trained on math data based on Qwen2.5-VL-7B [6]. Results from reasoning models post-trained with mathematical data are de-emphasized, while our ViGaL models are exclusively post-trained using visual games. Best scores of post-trained models in each "Avg." column are highlighted in **bold**.

Preserving general visual capabilities while reasoning enhancement. To comprehensively examine whether generalization on reasoning tasks leads to degradation in general visual capabilities, we evaluate ViGaL Snake + Rotation on a broader set of MLLM benchmarks. As shown in Tab. 4, compared to Qwen2.5-VL-7B prior to RL tuning, our model maintains comparable general visual performance while achieving stronger math reasoning results. In contrast, other models that improve math performance through RL post-training often exhibit substantial drops in general visual capabilities. These results demonstrate that our gameplay-based approach enables math generalization without compromising other visual abilities.

3.2. Ablation Study

We ablate key design choices in the Snake environment, evaluate each variant on downstream benchmarks, and report the results in Tab. 5 and Fig. 4. The corresponding ablation for the Rotation environment is provided in Appendix Sec. B.2.

Reasoning instructions in the text prompt help. We use reasoning instructions, such as "`finding the nearest apple by calculating Manhattan distances`", in the text prompts to guide the model thinking chains. The complete text prompts are in Appendix Sec. A.2. In Tab. 5a, we demonstrate that reasoning instructions brings significant improvement of 1.9%, from 59.5% to 61.4%, for Snake in average accuracy over the three out-of-domain benchmarks. Fig. 4a shows that integrating reasoning instructions during training significantly increases response length. These results highlight the effectiveness of adding reasoning instructions in the text prompt, helping RL training and generalization to downstream benchmarks.

Model		CLEVR+			Multi-Discipline		
	Avg.	Avg.	CLEVR-M	S-CLEVR	Avg.	MMMUval	MMMU-Prooverall
Proprietary Model							
GPT-4o [30]	55.9	51.2	68.1	34.3	60.5	69.1	51.9
Gemini-2.0-Flash [58]	–	46.3	64.9	27.6	–	71.9	–
General Multimodal Language Model							
InternVL2.5-8B [13]	54.8	64.4	93.5	35.3	45.2	56.0	34.3
Llava-OV-7B [36]	42.9	49.4	69.7	29.1	36.5	48.8	24.1
Qwen2.5-VL-7B [6]	50.3	54.9	74.6	35.2	45.7	54.3	37.0
Multimodal Reasoning Model Post-Trained on Qwen2.5-VL-7B							
R1-Onevision-7B [70]	53.7	65.1	75.5	54.7	42.3	51.9	32.6
R1-VL-7B [11]	53.9	68.0	87.4	48.6	39.7	50.0	29.4
MM-Eureka-Qwen-7B [48]	62.8	79.3	98.4	60.1	46.4	55.8	36.9
Reason-RFT-Zero-7B [57]	58.6	76.2	99.4	53.0	40.9	51.2	30.6
VLAA-Thinker-7B [10]	61.7	83.4	94.7	72.1	40.1	48.2	31.9
OpenVLThinker-7B [17]	60.4	82.4	93.8	71.0	38.5	54.8	22.1
ViGaL Snake	64.4	**82.6**	92.6	72.6	46.2	55.8	36.6
ViGaL Rotation	63.3	80.7	93.0	68.3	45.9	54.1	37.7
ViGaL Snake + Rotation	**64.7**	81.7	91.9	71.4	**47.7**	58.0	37.4

Table 3 | **Main results on multimodal spatial and multi-discipline reasoning benchmarks.** We extend our evaluation to non-mathematical reasoning tasks, comparing with multimodal reasoning models post-trained on domain-specific data based on Qwen2.5-VL-7B [6]. CLEVR-M denotes CLEVR-Math [41], and S-CLEVR stands for Super-CLEVR [40]. Results from reasoning models post-trained with corresponding in-domain data are de-emphasized, while our ViGaL models remain exclusively post-trained using visual games. Best scores of post-trained models in each "Avg." column are highlighted in **bold**.

Reward design of pre-text game matters for downstream tasks. We show that reward design of RL for games plays a crucial role for the downstream tasks. As shown in Tab. 5b, we first ask the model to predict only the best next move, defined as the action that moves toward the closest apple while avoiding death. In our improved reward design, we task the model with simultaneously predicting both the best and worst next moves, where the worst move leads directly to losing the game. As shown in Fig. 4b, predicting both best and worse moves improves the reasoning length, implying better thinking abilities. More importantly, it leads to improvements across all downstream tasks, bringing an average increase of 1.8%. These results suggest that proper reward design in pre-text game can improve not only gameplay capabilities but also generalization to downstream tasks.

Furthermore, inspired by several prior works that improve model performance without labeled rewards [75] or with random labels [54], we also provide a random reward ablation, where we still ask the model to predict both best and worst moves but use random moves as the labels. We report the results in the last row in Tab. 5b. In our gameplay setting, RL with random labels reports 49.4% on averagne and does no provide significant gains over the base model, different from the conclusions in prior works [54]. Potential explanations lie in the difference in data domains and base models, where other works applied random labels to text-only mathematical data while our work applies random labels to visual game data.

Model		General			Vision-Centric						OCR & Chart				
	Avg.	Avg.	Muir-Bench	CRPE$_{\text{rel.}}$	Avg.	MMVP	Real-WorldQA	MMStar	BLINK$_{\text{val}}$	MME$_{\text{p}}$	Avg.	AI2D$_{\text{w. M.}}$	SEED-Bench-2+	DocVQA val	OCR-Bench
Proprietary Model															
GPT-4o [30]	74.8	72.3	68.0	76.6	69.4	–	75.4	64.7	68.0	1614	82.6	84.6	72.0	91.1	736
General Multimodal Language Model															
Qwen2.5-VL-7B [6]	72.4	68.0	59.6	76.4	65.8	74.3	68.5	63.9	56.4	1698	83.3	83.9	70.4	95.7	864
Multimodal Reasoning Model Post-Trained on Qwen2.5-VL-7B															
R1-Onevision-7B [70]	–	66.8	46.3	87.3	56.5	61.3	58.0	57.8	48.7	1504	–	–	–	–	–
R1-VL-7B [11]	67.4	63.3	54.1	72.4	59.6	70.3	61.4	55.6	51.0	1657	79.2	81.7	66.4	89.4	81.0
MM-Eureka-Qwen-7B [48]	71.8	**68.9**	61.1	76.7	65.1	74.3	66.1	65.9	54.0	1626	81.5	84.3	68.2	92.0	87.0
Reason-RFT-Zero-7B [57]	68.4	66.9	58.5	75.2	58.5	58.0	65.3	59.1	51.6	1653	79.8	83.3	68.0	88.1	82.0
VLAA-Thinker-7B [10]	69.7	65.9	57.1	74.6	62.6	71.6	65.4	60.4	53.0	1593	80.6	83.4	67.4	90.9	84.5
OpenVLThinker-7B [17]	–	64.3	52.8	75.8	50.4	32.3	60.2	59.1	49.9	1513	–	–	–	–	–
ViGaL Snake + Rotation	**72.2**	68.6	60.5	76.7	**65.7**	74.6	67.3	65.4	55.6	1685	**82.2**	84.8	69.1	92.7	86.6

Table 4 | **Main results on multimodal language benchmarks targeting more general and comprehensive visual ability.** We compare with models post-trained on Qwen2.5-VL-7B [6]. Best category averages are highlighted in **bold**. Note that MME$_{\text{p}}$ is excluded from vision-centric category average accuracy due to scale differences.

Controlling game difficulty to steadily improve reasoning. Gameplay for RL post-training offers the unique opportunity to easily control the difficulty of the task itself. We present an ablation study on the importance of difficulty control. We define difficulty based on snake length, where states with longer snakes are considered more difficult. For our controlled difficulty approach, we collect training data using states where snake length falls within a moderate range between 1 and 5. As shown in Fig. 4c, models trained with the difficulty control strategy maintain relatively stable trends of increasingly longer responses throughout training. In contrast, models without difficulty control, which contain hard samples, experience struggle in gameplay. As shown in Tab. 5c, the approach with difficulty control achieves 61.4% overall accuracy compared to 60.6% without difficulty control. These findings suggest that our game engine can easily generate data with suitable difficulty to stabilize RL training and can help prevent model collapse during optimization.

RL on games shows data scalability. Thanks to using game engine, we can generate data at any scale with high flexibility. To show data scalability on RL of visual games, we conduct experiments using 16k and 32k snake game samples, respectively. As in Tab. 5d, scaling data from 16k to 32k brings a performance improvement of 1.3% on average across all domains. This suggests the potential of the proposed ViGaL paradigm to improve downstream performance by easily scaling training data, which contrasts with the data scaling challenges of domain-specific human annotated data, requiring extensive manual effort.

Both text and vision contribute to better visual reasoning. To isolate the contributions of text and vision modalities, we conduct an ablation study with a text-only setting. In this setup, we represent game states—including snake positions, apple locations, and boundary constraints—using only textual descriptions during RL training. The model trained with text-only inputs on the Snake game demonstrates substantial improvements across all multimodal benchmarks, with average performance increasing from 49.1% to 59.6%. Incorporating visual inputs yields an additional 1.8% performance gain. These results demonstrate that multimodal RL enhances visual reasoning capabilities, with complementary contributions from both text and vision modalities. Fig. 4d shows that including the vision modality leads to increased response length during RL training, suggesting more detailed reasoning processes.

RL generalizes better than SFT from games to math. To evaluate the out-of-domain generalization of ViGaL, we compare it with supervised fine-tuning (SFT) using identical visual game data. Tab. 5f shows that SFT with Snake game data degrades the base model's performance on both mathematical reasoning and geometry tasks by a notable 9.7% and 12.7%, respectively.

(a) Text prompt design.

prompt	Avg.	Math	CLEVR+	Geo.
base model	49.1	47.7	54.9	44.8
w/o reasoning instr.	59.5	48.0	80.4	50.1
w/ reasoning instr.	**62.3**	49.4	82.6	55.0

(b) Reward design.

reward	Avg.	Math	CLEVR+	Geo.
base model	49.1	47.7	54.9	44.8
best moves	59.6	48.2	80.4	50.2
best & worst moves	**62.3**	49.4	82.6	55.0
w/ random label	49.4	47.5	55.4	47.5

(c) Difficulty control.

difficulty control	Avg.	Math	CLEVR+	Geo.
base model	49.1	47.7	54.9	44.8
w/o difficulty control	60.6	48.8	81.4	51.8
w/ difficulty control	**62.3**	49.4	82.6	55.0

(d) Data scalability.

training samples	Avg.	Math	CLEVR+	Geo.
base model	49.1	47.7	54.9	44.8
16K	60.1	48.9	81.2	50.3
36K	**62.3**	49.4	82.6	55.0

(e) Input modality.

input modality	Avg.	Math	CLEVR+	Geo.
base model	49.1	47.7	54.9	44.8
text	59.6	48.5	80.1	50.3
vision & text	**62.3**	49.4	82.6	55.0

(f) SFT vs. RL.

post-training	Avg.	Math	CLEVR+	Geo.
base model	49.1	47.7	54.9	44.8
SFT	47.2	38.0	71.5	32.1
RL	**62.3**	49.4	82.6	55.0

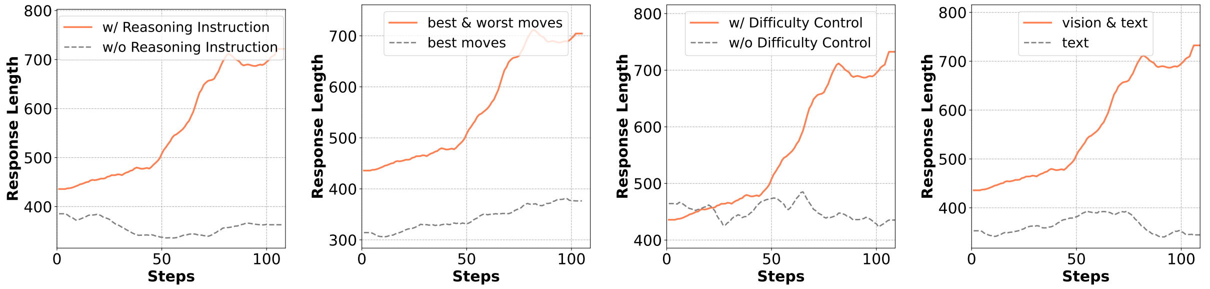
Table 5 | **Ablation study.** We ablate different aspects of ViGaL with Snake environment and evaluate on downstream benchmarks. The similar evaluation with Rotation is in Sec. B.2 in the Appendix. Each benchmark consists of several subtasks (Tab. 2 and Tab. 3), and we report their averages. The base model is Qwen2.5-VL-7B, whose results are in gray. The default settings in Tab. 2 and Tab. 3 are highlighted in blue.

While SFT produces modest improvements on CLEVR+, these gains are substantially smaller than those achieved by RL. Overall, RL improves performance by 12.3%, whereas SFT decreases performance by 1.9%. This stark contrast demonstrates that RL better preserves and extends the model's reasoning capabilities to new domains.

Different games benefit distinct math subfields. We hypothesize that gameplay fosters fundamental skills like spatial modeling and sequential planning that can transfer to visual math questions. Different games may enhance distinct reasoning abilities. To investigate this hypothesis, we analyze accuracy differences across MathVerse [74] subcategories between ViGaL models trained with Snake or Rotation, as shown in Fig.5. Training on the Snake game significantly improves performance on the Expressions and Coordinates subcategories. Both tasks involve algebraic functions and coordinate-level interpretations of graphical representations, closely aligning with the spatial reasoning in Snake's 2D-grid environment. In contrast, training on Rotation notably enhances performance on questions about angles and lengths, consistent with Rotation's requirement to reason about rotational angles of 3D objects. These results suggest that different games obtains specialized reasoning skills that correspond to their unique gameplay mechanics. Furthermore, joint training on both games leads to improvements across *all* reasoning categories (see Appendix Sec.B.4). We also include qualitative analyses illustrating improvements in mathematical reasoning after RL in Appendix Sec. C.

4. Related Work

Reinforcement Learning in MLLMs. Reinforcement Learning (RL) increasingly enhances reasoning in Large Language Models (LLMs) beyond Supervised Fine-Tuning (SFT). Text-only models like DeepSeek-R1 [16] show RL's efficacy, especially with rule-based rewards, for complex reasoning. This paradigm is now actively being extended to Multimodal LLMs (MLLMs).



(a) Text prompt.

(b) Reward design.

(c) Difficulty control.

(d) Input modality.

Figure 4 | **Ablation study on the impact of design choices on response length.** Solid orange lines represent the full configuration, while dashed grey lines indicate the ablated counterpart. The graphs demonstrate how (a) incorporating reasoning instructions, (b) designing rewards to consider both best and worst moves, (c) implementing difficulty control, and (d) utilizing multimodal inputs all contribute to increased response length as training progresses, implying better reasoning abilities.

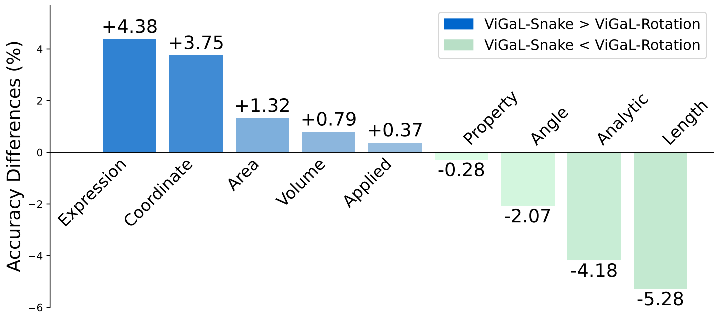


Figure 5 | **Snake *vs.* Rotation: Subfield differences on MathVerse.** Positive values indicate better results from ViGaL Snake, and negative values measure how much ViGaL Rotation performs better. Interestingly, Snake enhances most on Expressions and Coordinates, tasks aligned with Snake's 2D grid. Rotation improves angle and length reasoning, reflecting its focus on 3D object rotations.

Recent MLLM research explores RL for improved visual reasoning, drawing inspiration from LLM successes. For instance, various works [11, 29, 50] investigate multi-stage training, trace supervision, or rule-based RL for specific visual subdomains like geometry and counting. Others focus on different RL algorithms like Process Reward Models (PRMs) [46, 69], often moving beyond SFT-based Chain-of-Thought generation [19, 60]. Many efforts are moving towards simpler rule-based rewards [29, 76] over complex reward models prone to hacking [21]. Unlike approaches that train on costly, domain-specific reasoning datasets, our ViGaL paradigm contributes by extending rule-based RL to simple, synthetic visual games, demonstrating that these can serve as scalable, cost-effective pre-text tasks.

Generalization in MLLMs. Achieving robust generalization to novel tasks, distributions, and domains is a central goal in the development of MLLMs. RL has shown promise for better out-of-distribution (OOD) generalization compared to SFT [11, 48], and developing multi-step reasoning like CoT [67] is itself a form of generalization. Generalization is often pursued by training on large, diverse instruction-following datasets [13, 38, 42] or by explicitly training general reasoning capabilities [29, 70]. While these methods advance OOD generalization, they typically operate within the same broad domain of complex visual reasoning as the training data. Our ViGaL paradigm, however, investigates a stronger form of out-of-domain generalization. We show fundamental skills learned from simple synthetic games transfer zero-shot to enhance performance on entirely different, complex domains like visual mathematics and multi-displine questions, without any exposure to corresponding domain-specific data.

5. Conclusion

We introduced Visual Game Learning (ViGaL), a novel post-training paradigm where MLLMs learn transferable reasoning by playing simple arcade-style games. Our core finding is that RL on games like Snake and Rotation, *without any in-domain math data*, significantly boosts MLLM performance on mathematical and multi-discipline benchmarks, surpassing specialized models and even large proprietary systems. Ablations confirm the importance of game design, reward structure, and that RL outperforms SFT, while distinct games unlock different skills. We posit that games instill fundamental cognitive primitives, suggesting a new avenue for using scalable, controllable synthetic games as powerful pre-text tasks to unlock generalizable reasoning. This work opens doors to exploring a broader range of game-based learning for robust AI. Future directions include investigating the synergistic effects among diverse games and developing a deeper understanding of the transfer mechanisms.

Acknowledgment We thank Haoqin Tu and Yuxuan Cheng for their valuable feedback on this manuscript. We also thank the authors of the paper Cambrian-1 [61] for the template reference.

References

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *ACL*, 2024.

[2] Fahad Alhasoun and Sarah Alneghiemish. Probabilistic programming bots in intuitive physics game play. In *AAAI*, 2021.

[3] Kelsey Allen, Franziska Brändle, Matthew Botvinick, Judith E. Fan, Samuel J. Gershman, Alison Gopnik, et al. Using games to understand the mind. *Nature Human Behaviour*, 2024.

[4] Kelsey R Allen, Kevin A Smith, and Joshua B Tenenbaum. Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *PNAS*, 2020.

[5] Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning-oriented reinforcement learning. *arXiv:2504.03380*, 2025.

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

[7] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *ICLR*, 2019.

[8] Lara Bertram. Digital learning games for mathematics and computer science education: The need for preregistered rcts, standardized methodology, and advanced technology. *Frontiers in Psychology*, 2020.

[9] Franziska Brändle, Kelsey R Allen, Josh Tenenbaum, and Eric Schulz. Using games to understand intelligence. In *CogSci*, 2021.

[10] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, et al. SFT or RL? an early investigation into training R1-Like reasoning large vision-language models. *arXiv:2504.11468*, 2025.

[11] Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-V: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V, 2025.

[12] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv:2403.20330*, 2024.

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024.

[14] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, et al. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv:2501.17161*, 2025.

[15] Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. *arXiv:2504.02546*, 2025.

[16] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025.

[17] Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Open-VLThinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv:2503.17352*, 2025.

[18] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[19] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-V: Exploring long-chain visual reasoning with multimodal large language models. *arXiv:2411.14432*, 2024.

[20] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multi-agent debate. In *ICML*, 2023.

[21] Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv:2312.09244*, 2023.

[22] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv:2306.13394*, 2023.

[23] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

[24] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, et al. G-LLaVA: Solving geometric problem with multi-modal large language model. *arXiv:2312.11370*, 2023.

[25] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *arXiv:2210.10760*, 2022.

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv:1803.07728*, 2018.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[28] Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. Open-RLHF: An easy-to-use, scalable and high-performance RLHF framework. *arXiv:2405.11143*, 2024.

[29] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, et al. Vision-R1: Incentivizing reasoning capability in multimodal large language models. *arXiv:2503.06749*, 2025.

[30] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, et al. GPT-4o system card. *arXiv:2410.21276*, 2024.

[31] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[32] Greg Kamradt. Snake Bench: Competitive snake game simulation with LLMs. https://github.com/gkamradt/SnakeBench, 2025.

[33] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[34] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

[35] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Workshop on Deep Reinforcement Learning Meets Structured Prediction*, 2019.

[36] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv:2408.03326*, 2024.

[37] Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024.

[38] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, et al. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv:2407.07895*, 2024.

[39] Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, et al. A survey on benchmarks of multimodal large language models. *arXiv preprint arXiv:2408.08632*, 2024.

[40] Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan L. Yuille. Super-CLEVR: A virtual benchmark to diagnose domain robustness in visual reasoning. In *CVPR*, 2023.

[41] Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-Math: A dataset for compositional language, visual and mathematical reasoning. In *IJCLR*, 2022.

[42] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/, 2024.

[43] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: On the hidden mystery of ocr in large multimodal models. *arXiv:2305.07895*, 2023.

[44] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, et al. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv:2310.02255*, 2024.

[45] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv:2105.04165*, 2021.

[46] Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyao Yu, Xinzhe Ni, Zicheng Lin, et al. URSA: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv:2501.04686*, 2025.

[47] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.

[48] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, et al. MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv:2503.07365*, 2025.

[49] OpenAI. Introducing OpenAI o1. https://openai.com/o1/, 2024.

[50] YingZhe Peng, Gongrui Zhang, Xin Geng, and Xu Yang. LMM-R1. https://github.com/TideDra/lmm-r1, 2025.

[51] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[52] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv:2205.06175*, 2022.

[53] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[54] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr. https://rethink-rlvr.notion.site/Spurious-Rewards-Rethinking-Training-Signals-in-RLVR-1f4df34dac1880948858f95aeb88872f, 2025. Notion Blog.

[55] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-LLaVA: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv:2406.17294*, 2024.

[56] Parshin Shojaee*†, Iman Mirzadeh*, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025.

[57] Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-RFT: Reinforcement fine-tuning for visual reasoning. *arXiv:2503.20752*, 2025.

[58] Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

[59] Tencent Hunyuan3D Team. Hunyuan3D 2.0: Scaling diffusion models for high-resolution textured 3d assets generation. *arXiv:2501.12202*, 2025.

[60] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, et al. LLaMAV-o1: Rethinking step-by-step visual reasoning in LLMs. *arXiv:2501.06186*, 2025.

[61] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

[62] Shengbang Tong, Zhuang Liu, Yuexiang Zhu, Xingjian Chen, Ruoyu Zhang, Bo Li, et al. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

[63] Bas Van Opheusden, Ionatan Kuperwajs, Gianni Galbiati, Zahy Bnaya, Yunqi Li, and Wei Ji Ma. Expertise increases planning depth in human gameplay. *Nature*, 2023.

[64] Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024.

[65] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with Math-Vision dataset. In *NeurIPS*, 2024.

[66] Nicholas R. Waytowich, Devin White, M.D. Sunbeam, and Vinicius G. Goecks. Atari-GPT: Investigating the capabilities of multimodal large language models as low-level policies for atari games. *arXiv:2408.15950*, 2024.

[67] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[68] X.AI. Grok-1.5 vision preview. https://x.ai/blog/grok-1.5v, 2024.

[69] Kun Xiang, Zhili Liu, Zihao Jiang, Yunshuang Nie, Runhui Huang, Haoxiang Fan, et al. AtomThink: A slow thinking framework for multimodal mathematical reasoning. *arXiv:2411.11930*, 2024.

[70] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, et al. R1-OneVision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv:2503.10615*, 2025.

[71] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[72] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024.

[73] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv:2504.13837*, 2025.

[74] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, et al. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? *arXiv:2403.14624*, 2024.

[75] Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

[76] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-Zero's "aha moment" in visual reasoning on a 2b non-sft model. *arXiv:2503.05132*, 2025.

Appendix

Content

A. Data

A.1. Training Data Synthesis

This section provides additional implementation details for the game training data synthesis process, expanding on the methodology outlined in Section 2.3.

For the Snake game, the environment consists of a 10×10 grid game board with two snakes of 1-grid initial length. At each time step t, each snake receives one action respectively to move, resulting in a new game state s_{t+1}.

To generate meaningful moves that accomplish the objective of collecting more apples while remaining alive, we implement a policy network based on Proximal Policy Optimization (PPO) [53]. The observation space is represented as a 10×10 grid with distinct values indicating empty cells (0), apples (1), the agent's own body (2), and other agents' bodies (3). These observations are stacked across 4 time steps to incorporate temporal information, resulting in an input tensor $\mathbf{X} \in \mathbb{R}^{10\times10\times4}$.

The policy network architecture consists of two convolutional layers with 3×3 kernels, followed by fully connected layers. The first convolution has $C_1 = 16$ output channels while the second has $C_2 = 32$ output channels, both followed by ReLU activation functions. After flattening, the features pass through a fully connected layer with 256 units before outputting action logits for the four possible movements (right, left, up, down), which are then transformed into a probability distribution $\boldsymbol{\pi}(a|s)$ using softmax. The value function follows a similar architecture but produces a single scalar output $V(s)$.

To prevent the snake from easily dying, we incorporate action priors that discourage suicidal moves by masking logits for dangerous actions (e.g., moving into walls or other snake bodies). The model employs the standard PPO objective with entropy regularization coefficient $\beta = 0.01$ to encourage exploration, along with a value function coefficient $\lambda = 0.5$ and a clipping parameter $\varepsilon = 0.2$. During training, we use a buffer size of 2048 and minibatch size of 32 with Adam optimizer at a learning rate $\eta = 10^{-3}$.

Agents receive a reward of $r = +1$ for collecting apples and a penalty of $r = -1$ for dying. This reward structure, combined with the PPO algorithm, enables agents to learn complex behaviors such as obstacle avoidance, apple pursuit, and multi-step trajectory planning. With this policy network, we continuously collect data from the Snake game, generating diverse game scenarios that serve as training examples for downstream RL training.

For the Rotation game, training data comprises synthetically generated visual puzzles focused on 3D spatial reasoning, specifically the understanding of object rotations. We utilized a diverse collection of 540 unique 3D object meshes in total, with 408 meshes sourced from Hunyuan3D 2.0 [59] and an additional 132 meshes from Hunyuan3D 2.5. Hunyuan3D is a large-scale 3D asset generation system capable of producing high-resolution textured objects, providing a wide variety of shapes and textures for our game. Our custom data generation pipeline produced pairs of images $(I_{\text{init}}, I_{\text{rot}})$ for each mesh, representing the object before and after a defined rotation.

The generation of each $(I_{\text{init}}, I_{\text{rot}})$ pair followed a precise sequence. First, to establish a diverse initial viewpoint for I_{init}, the 3D object was subjected to a base orientation: 0° rotation around its x-axis, an angle selected from the set $\{0°, 45°, \ldots, 315°\}$ around its y-axis, and 0° around its z-axis. To further enhance visual variety and prevent the learning of trivial transformations from canonical poses, an additional z-axis rotation, chosen from $\{0°, 30°, \ldots, 330°\}$, was subsequently applied. The rendering of the object after these compound initial transformations yielded the I_{init} image. Subsequently, the I_{rot} image was generated by applying the target rotation to the object state depicted in I_{init}. This target rotation was exclusively around the z-axis by an angle of either 90° or 180°, which also served as the ground truth label for the sample. Our coordinate system is defined with the x-axis pointing to the right, the y-axis pointing upward, and the z-axis pointing outward from the screen toward the viewer; thus, all target rotations occur in the plane of the image.

All objects were rendered at a 512×512 pixel resolution using a consistent perspective camera providing a frontal view, under standardized lighting conditions. Visualizations of coordinate axes were not included in the rendered images. This process resulted in approximately 32k unique $(I_{\text{init}}, I_{\text{rot}})$ pairs derived from the pool of 537 meshes allocated for generating test instances. As detailed in Section 2.1, each training instance presented to the MLLM comprised four images—an example pair $(I_{\text{init}}^{\text{ex}}, I_{\text{rot}}^{\text{ex}})$ and a task pair $(I_{\text{init}}^{\text{task}}, I_{\text{rot}}^{\text{task}})$. The example pairs were generated using a separate, dedicated set of 3 meshes, ensuring that the objects seen in in-context examples were distinct from those used in the test portion of any given prompt. Both example and test pairs were generated via the methodology described above.

A.2. Training Prompt in Visual Game Learning

Prompt for Snake Game

Your role is to guide a snake within a Snake game featuring multiple apples.

This game is played on a board of size 10 by 10. The board uses a standard Cartesian coordinate system, where (0,0) represents the bottom-left position and (9,9) is the top-rightmost coordinate.

Apples at: {apple_position}

Direction of Your Last Action: {last_action}

Rules:
1) If you move onto an apple, you grow and gain 1 point.
2) If your head moves to a position where its coordinates (x, y) are outside the board boundaries (meaning x < 0, x > 9, y < 0, or y > 9), or into a space occupied by another snake's body, or into a space occupied by your own body, you die. That's the worst move.

3) The goal is to prioritize snake not die, then efficiently collecting apples. First avoid the worst move, then for each apple, find the nearest apple by calculating Manhattan distances. But only choose best next move to get closer the nearest apple if you can confirm best next move will not run outside the range of the listed coordinates, run into the position of another snake, or yourself. Otherwise it will be the worst move.

Your snake with the ID {snake_id} in {snake_color} has its head now positioned at {snake_position}, and its body extends to {body_position} You should avoid your next move into your own snake's position.

Enemy snakes in {enemy_color} positions: {enemy_position}.

Decreasing your x coordinate is to the LEFT, increasing your x coordinate is to the RIGHT.
Decreasing your y coordinate is DOWN, increasing your y coordinate is UP.

Read out another snake's position and apple position. Try to predict another snake's next move and avoid colliding with it.

Best answer is one of next move that is the closest to the apple and not lead to your death. Worst answer is all of next moves 1. makes your head's coordinates (x, y) are outside the board boundaries, meaning x < 0, x > 9, y < 0, or y > 9. 2. moves into a position occupied by another snake's body. 3. moves into a position occupied by body of yourself.

Check all the next moves to list out all the worst moves in `<worst_answer>` tag. If no worst answer, return None for worst answer, e.g., "`<worst_answer>None</worst_answer>`"

The best answer and the worst answer are mutually exclusive and different.

You need first to give your reasoning process then to choose one of best next move and worst next move from ['UP', 'DOWN', 'LEFT', 'RIGHT'].

The reasoning process and answer are enclosed within <think> </think>, <best_answer> </best_answer> and <worst_answer> </worst_answer> tags, respectively, i.e., "<think> reasoning process here </think><best_answer> one best move here </best_answer><worst_answer> all worst moves here </worst_answer>"

Prompt for Rotation Game

I'm showing you 4 images. Images 1-2 are an example pair, and Images 3-4 are the test pair. In each pair, the first image shows the initial orientation, and the second shows the object after rotation.

EXAMPLE OF ROTATION

Example: Image 1 shows the initial view and Image 2 shows the object after a 180 degree rotation.

YOUR TASK
Now, considering the transformation from Image 3 (initial) to Image 4 (rotated)
. Determine the angle of rotation from Image 3 to Image 4 on the plane
Analyze the rotation carefully using the example pair (Images 1-2) as a reference.

1. Coordinate System Transformation:
- Draw an x-y coordinate system on both original and rotated images with origin at center
- Identify a distinct feature point and note its coordinates in both images
- Apply rotation matrix equations to verify the transformation

Example: A star icon at coordinates (3,1) in the original image appears at (-1,3) in the rotated image. Testing with the 90° clockwise rotation matrix [cos(90°), sin(90°); -sin(90°), cos(90°)] confirms the transformation from (3,1) to (-1,3), verifying a 90° clockwise rotation.

2. Angular Displacement Measurement:
- Mark the image center as the origin in both images
- Draw a straight line from center to a distinctive feature in both images
- Measure the angle between these two lines using counterclockwise as positive

Example: A line from center to a red dot makes a 30° angle with horizontal in the original image. In the rotated image, this line makes a 210° angle with horizontal. The difference (180°) indicates a clockwise 180° rotation.

3. Symmetry Axis Tracking:
- Identify major symmetry axes in the original image
- Locate the same symmetry axes in the rotated image
- Calculate the angular displacement between original and rotated axes

Example: A rectangular logo has vertical and horizontal symmetry axes. After rotation, the vertical axis now points right and horizontal points down. This 90° shift of both axes confirms a clockwise 90° rotation.

4. Triangle Configuration Analysis:
- Select three non-collinear distinct points forming a triangle in both images
- Compare the orientation of this triangle in both images using vector cross products
- Determine rotation angle from the triangle's orientation change

Example: Three points form a right triangle with vertices clockwise arranged. After rotation, the same triangle has its vertices arranged in counterclockwise order while maintaining the same shape. This inversion indicates a clockwise 180° rotation.

5. Polar Coordinate Comparison:
- Convert key points to polar coordinates (r,θ) relative to image center
- Compare θ values of the same features in original and rotated images
- Calculate consistent angular difference across multiple points

Example: A feature at polar angle 45° in the original image appears at 135° in the rotated image. Another feature shifts from 10° to 100°. Both show a +90° shift in polar angle, confirming a clockwise 90° rotation.

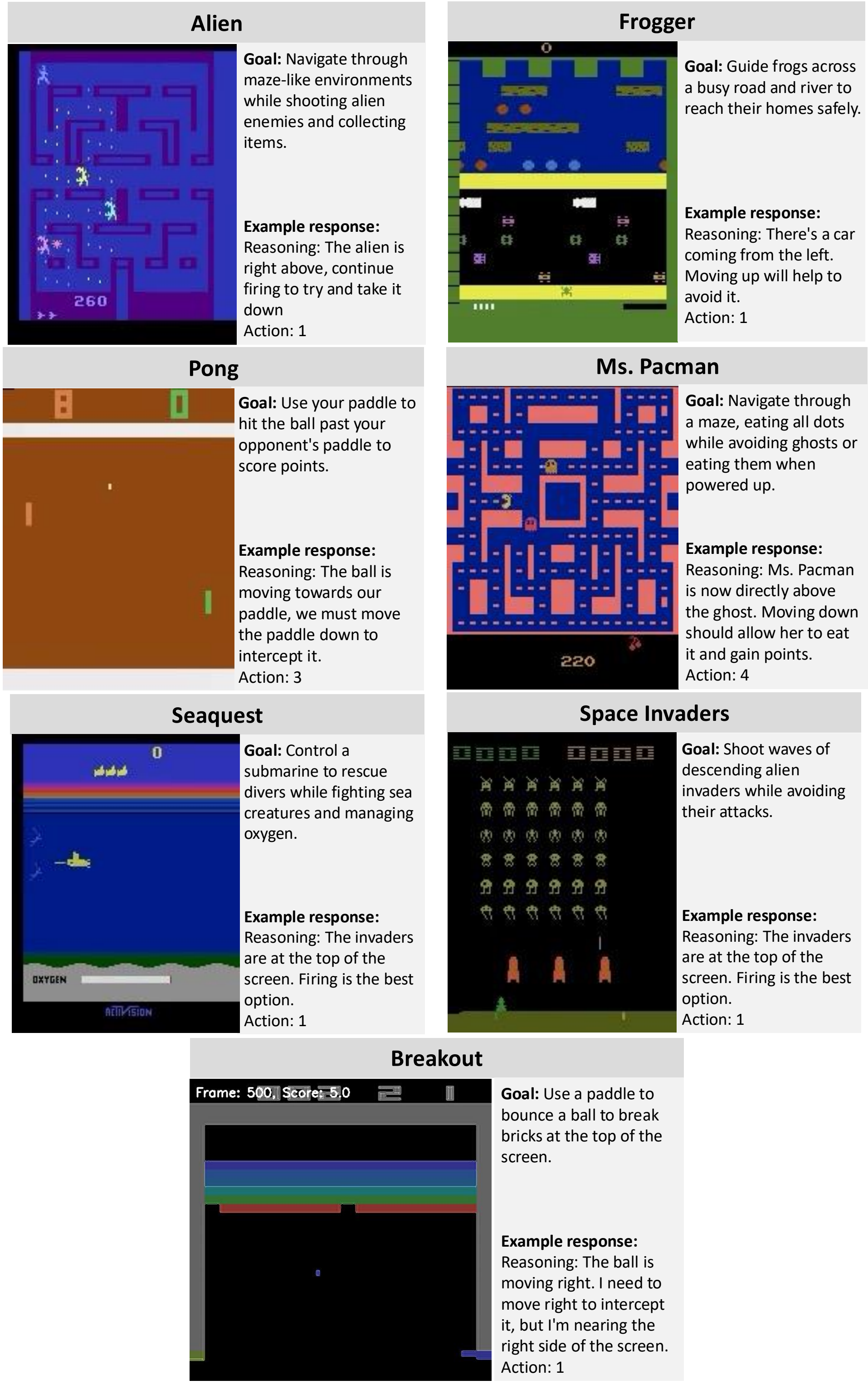
Choose the rotation angle from this list: ['counter clockwise 90', '180']

The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., "<think> reasoning process here </think><answer> answer here </answer>"

We provide additional details regarding the training prompts used in RL training with the Snake and Rotation games. Our game prompts consist of two main parts: game settings and reasoning instructions. For the Rotation game, to promote diverse problem-solving approaches, each training instance included one spatial reasoning strategy hint randomly selected from a predefined set of five: 1. Coordinate System Transformation, 2. Angular Displacement Measurement, 3. Symmetry Axis Tracking, 4. Triangle Configuration Analysis, or 5. Polar Coordinate Comparison. **Bold text** indicates reasoning instructions synthesized by GPT-4o [30].

B. Evaluation

B.1. Evaluation Detail of Atari Game



Alien

Goal: Navigate through maze-like environments while shooting alien enemies and collecting items.

Example response:
Reasoning: The alien is right above, continue firing to try and take it down
Action: 1

Frogger

Goal: Guide frogs across a busy road and river to reach their homes safely.

Example response:
Reasoning: There's a car coming from the left. Moving up will help to avoid it.
Action: 1

Pong

Goal: Use your paddle to hit the ball past your opponent's paddle to score points.

Example response:
Reasoning: The ball is moving towards our paddle, we must move the paddle down to intercept it.
Action: 3

Ms. Pacman

Goal: Navigate through a maze, eating all dots while avoiding ghosts or eating them when powered up.

Example response:
Reasoning: Ms. Pacman is now directly above the ghost. Moving down should allow her to eat it and gain points.
Action: 4

Seaquest

Goal: Control a submarine to rescue divers while fighting sea creatures and managing oxygen.

Example response:
Reasoning: The invaders are at the top of the screen. Firing is the best option.
Action: 1

Space Invaders

Goal: Shoot waves of descending alien invaders while avoiding their attacks.

Example response:
Reasoning: The invaders are at the top of the screen. Firing is the best option.
Action: 1

Breakout

Goal: Use a paddle to bounce a ball to break bricks at the top of the screen.

Example response:
Reasoning: The ball is moving right. I need to move right to intercept it, but I'm nearing the right side of the screen.
Action: 1

Figure 6 | **Goal and example response from model of Atari games used for evaluation.** We implement 7 kinds of Atari games from Atari-GPT [66].

To evaluate out-of-distribution generalization, we test ViGaL on Atari-GPT [66], a benchmark for evaluating MLLMs as decision-making agents in Atari video games, as shown in Fig. 6. The benchmark consists of seven different Atari games: Alien, Frogger, Pong, Ms. Pacman, Seaquest, Space Invaders and Breakout. These games present diverse visual environments which is different from Snake game and Rotation game, and require different strategic approaches to finish the goal, making them an ideal test bed for ViGaL evaluating out-of-distribution generalization capabilities.

For evaluation, we input game frames as pixel observations to our model, following the established protocol in Atari-GPT. Specifically, each game frame is resized from 210 × 160 × 3 to 512 × 512 × 3, then provided to our model along with game-specific action information. We maintain a context buffer containing the two previous frames and responses together with the current frame to enable temporal reasoning. Following Atari-GPT, we implement frame skipping of 8 frames, which extends the standard 4-frame skipping in ALE to reduce computational intensity while preserving gameplay continuity.

We evaluate our method through four independent rollouts of 1,000 timesteps each and report the average cumulative reward, with results presented in Tab. 1c.

B.2. Ablation On Rotation Game

Table 6 | **Ablation study.** Similar to the evaluation in Tab. 5, we analyze how different aspects of our post-training strategy within the Rotation game affect downstream generalization benchmarks. The base model is Qwen2.5-VL-7B, with results shown in gray. The default settings from Tab. 2 and Tab. 3 are highlighted in blue. We observe the same improvement trends for each strategy as reported in Tab. 5.

(a) Prompt design.

prompt	**Avg.**	**Math**	**CLEVR+**	**Geo.**
base model	49.1	47.7	54.9	44.8
w/o Reasoning Instruction	61.4	48.9	80.4	54.8
w/ Reasoning Instruction	62.6	49.3	80.7	57.9

(b) SFT vs. RL.

post-training	**Avg.**	**Math**	**CLEVR+**	**Geo.**
base model	49.1	47.7	54.9	44.8
SFT	55.6	44.0	75.4	47.5
RL	62.6	49.3	80.7	57.9

(c) Difficulty control.

difficulty control	**Avg.**	**Math**	**CLEVR+**	**Geo.**
base model	49.1	47.7	54.9	44.8
w/o difficulty control	61.0	48.0	80.2	54.8
w/ difficulty control	62.6	49.3	80.7	57.9

As shown in Tab. 6, we conduct a similar ablation study to Tab. 5, but replace the Snake game environment with the Rotation game. Our results demonstrate the same consistent improvement trends on downstream generalization benchmarks for each strategy employed.

Specifically, we control the task difficulty by varying the rotation angles between two images. In the uncontrolled difficulty setting, the rotation angle between images can be clockwise 90°, counter-clockwise 90°, or 180°. However, we found that explicitly requiring the model to distinguish between clockwise and counter-clockwise rotations leads to training difficulties. Therefore, we remove it and only retain option of clockwise 90° and 180° rotations.

Unlike the Snake game, we cannot conduct the ablations shown in Tab. 5e because the Rotation game is inherently vision-dependent and requires visual input. Similarly, we cannot perform the ablations in Tab. 5b because the Rotation game provides only binary answer options, making it impossible to meaningfully designate both "best" and "worst" answers simultaneously.

B.3. Synergistic Integration with Mathematical Data

Model	Math Avg.	MathVista	MathVerse	MathVision
base model	47.7	68.0	49.0	26.0
MM-Eureka-Qwen-7B	50.1	73.0	50.3	26.9
ViGaL (w/o Math Data)	50.6	71.9	52.4	27.5
ViGaL (w/ Math Data)	51.8	72.3	54.5	27.7

Table 7 | **Ablation study on Math data.** We conduct experiment of additionally training ViGaL on mathematical data MMK12 [48]. The base model is Qwen2.5-VL-7B, whose results are in gray. The setting with highest average accuracy is highlighted in blue .

Although our work primarily demonstrates mathematical performance improvement without training on math data, we conducted additional experiments to explore the synergistic benefits of integrating mathematical data into our training pipeline. In our experimental setup, we implemented a two-stage training process. In stage 1, we followed our original approach, training the model exclusively on Snake and Rotation games. For stage 2, we trained our model on MMK12 [48], a multimodal mathematical reasoning dataset containing approximately 12k examples. We maintained identical data and training settings as MM-Eureka-Qwen-7B [48]. The only difference was our model's additional stage 1 training on visual games.

As shown in Tab. 7, the integration of mathematical data in stage 2 yielded a continuous improvement of 0.9% on average across three mathematical benchmarks compared to using only stage 1 training. This demonstrates the synergistic relationship between our visual game learning approach and mathematical data fine-tuning. Moreover, ViGaL (w/ Math Data) significantly outperformed MM-Eureka-Qwen-7B by 1.4% on mathematical benchmarks on average, despite both models using same math data. These results suggest that visual game learning can serve as an effective foundation training stage that can be further enhanced with domain-specific data to improve performance on target tasks.

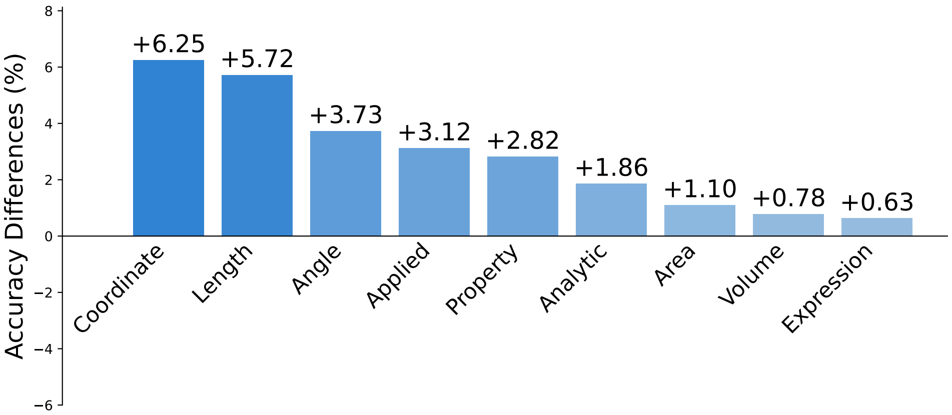


Figure 7 | **Accuracy differences between ViGaL-Snake+Rotation and base model without RL training across mathematical subfields in Mathverse.** The synergistic effects of jointly training on two games observed suggest that complementary games can enhance overall mathematical reasoning capabilities.

B.4. Synergistic Effects of Multi-Game Training

As discussed in Sec. 3.2, our analysis reveals that each game develops distinct reasoning abilities in the model. To investigate potential combined benefits, we conducted experiments where models were trained simultaneously on *both* the Snake and Rotation games. Fig. 7 shows that joint training effectively combines the strengths of each individual game, improving performance across the mathematical areas where each game shows particular effectiveness, resulting in greater overall gains on Mathverse. These results suggest that strategically combining games with complementary strengths offers a simple yet effective approach to enhance model generalization abilities.

B.5. Reasoning Ability Boundary via Pass@k Evaluation

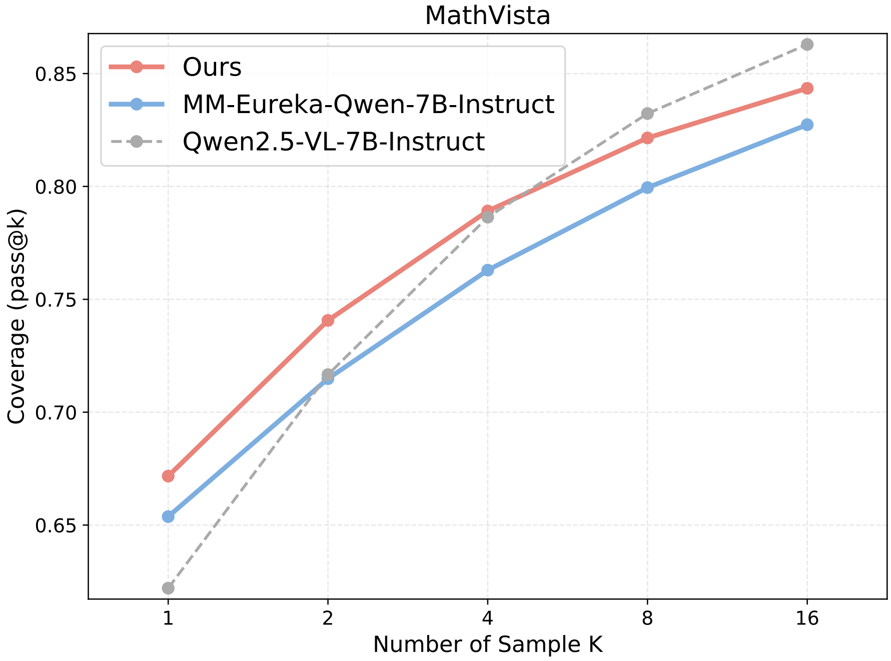


Figure 8 | Pass@k performance curves on MathVista comparing base models with their zero-RL counterparts trained on mathematical data and game data, respectively.

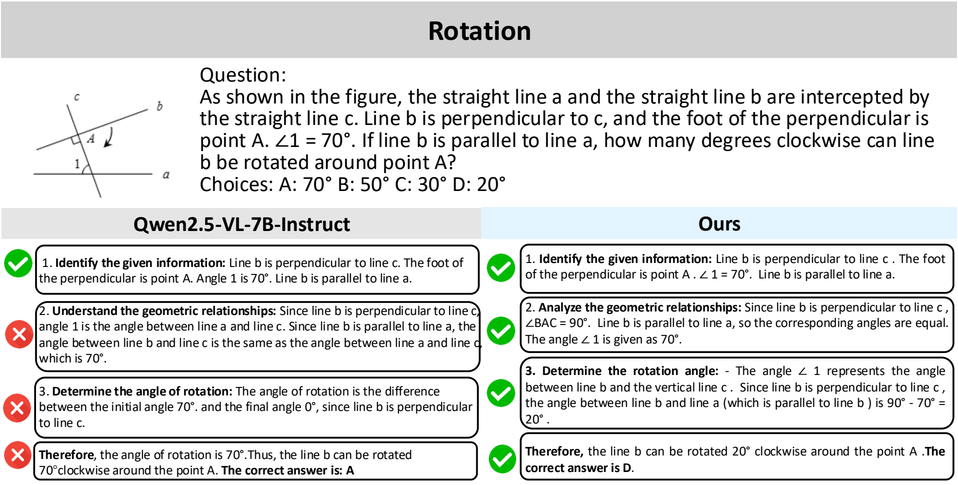
We explore the reasoning ability boundary of models trained with different RL approaches by evaluating the pass@k metric. This metric measures the probability that at least one of k independent model samples solves a given problem, indicating the true scope or boundary of a model's reasoning capability - essentially what problems the model can potentially solve given enough sampling attempts.

We evaluate the pass@k performance of three models: the Base Model without RL training, MM-Eureka-Qwen-7B-Instruct, and our ViGaL. As shown in Fig. 8, our ViGaL consistently demonstrates increasing pass@k scores on Mathverse as k increases. This finding suggests that our approach can effectively solve complex problems when allowed multiple reasoning attempts, uncovering capabilities not apparent in single-sample evaluations.

Moreover, compared to the other RL-trained model, MM-Eureka-Qwen-7B-Instruct, our model achieves a steeper improvement in pass@k as k increases. This indicates that ViGaL possesses a broader reasoning boundary and stronger reasoning abilities, enabling it to solve a wider range of problems when given sufficient opportunities to explore different solution paths.

Finally, our results demonstrate that as k increases, base models without RL training eventually outperform RL-trained models. This aligns with the findings in [73] that highlight a fundamental limitation of reinforcement learning with verifiable rewards (RLVR): while RL training significantly improves performance at small k values (e.g., pass@1), base models possess a wider coverage of solvable problems. This suggests a trade-off where RL optimization focuses on solving high-probability problems at the expense of broader solution coverage. Future work should explore RLVR algorithms that can improve pass@k performance across all values of k, effectively extending the reasoning boundary beyond that of the base model.

C. Case Study



Rotation

Question:
As shown in the figure, the straight line a and the straight line b are intercepted by the straight line c. Line b is perpendicular to c, and the foot of the perpendicular is point A. ∠1 = 70°. If line b is parallel to line a, how many degrees clockwise can line b be rotated around point A?
Choices: A: 70° B: 50° C: 30° D: 20°

Qwen2.5-VL-7B-Instruct

✓ 1. **Identify the given information:** Line b is perpendicular to line c. The foot of the perpendicular is point A. Angle 1 is 70°. Line b is parallel to line a.

✗ 2. **Understand the geometric relationships:** Since line b is perpendicular to line c, angle 1 is the angle between line a and line c. Since line b is parallel to line a, the angle between line b and line c is the same as the angle between line a and line c, which is 70°.

✗ 3. **Determine the angle of rotation:** The angle of rotation is the difference between the initial angle 70°. and the final angle 0°, since line b is perpendicular to line c.

✗ **Therefore**, the angle of rotation is 70°.Thus, the line b can be rotated 70°clockwise around the point A. **The correct answer is: A**

Ours

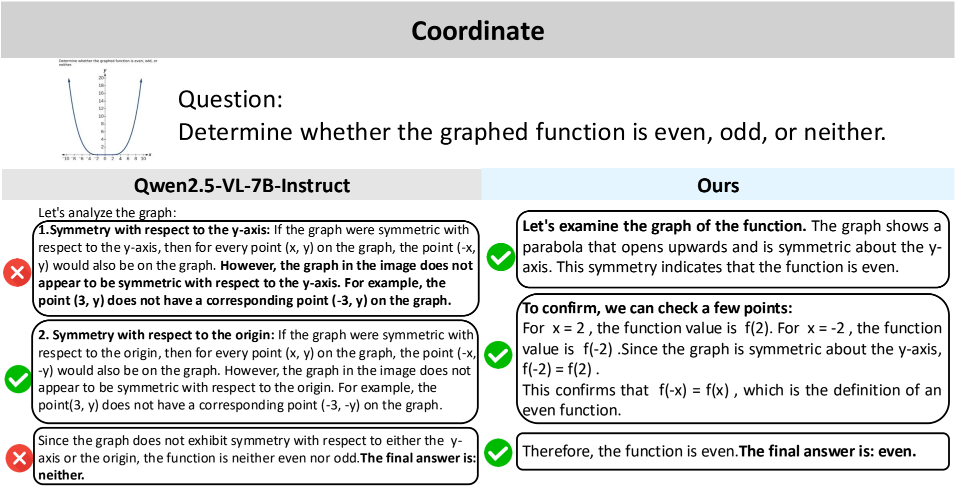
✓ 1. **Identify the given information:** Line b is perpendicular to line c . The foot of the perpendicular is point A . ∠ 1 = 70°. Line b is parallel to line a.

✓ 2. **Analyze the geometric relationships:** Since line b is perpendicular to line c , ∠BAC = 90°. Line b is parallel to line a, so the corresponding angles are equal. The angle ∠ 1 is given as 70°.

✓ **3. Determine the rotation angle:** - The angle ∠ 1 represents the angle between line b and the vertical line c . Since line b is perpendicular to line c , the angle between line b and line a (which is parallel to line b) is 90° - 70° = 20° .

✓ **Therefore,** the line b can be rotated 20° clockwise around the point A .**The correct answer is D**.

(a) A case study from Mathverse. Base model misinterpreted the geometric configuration and rotation direction, while our model correctly identified the perpendicular relationship and calculated the proper angle.


Determine whether the graphed function is even, odd, or neither
20
18
16
14
12
10
8
6
4
2
-10 -8 -6 -4 -2 0 2 4 6 8 10
x

Coordinate

Question:
Determine whether the graphed function is even, odd, or neither.

Qwen2.5-VL-7B-Instruct

Let's analyze the graph:

✗ **1.Symmetry with respect to the y-axis:** If the graph were symmetric with respect to the y-axis, then for every point (x, y) on the graph, the point (-x, y) would also be on the graph. **However, the graph in the image does not appear to be symmetric with respect to the y-axis. For example, the point (3, y) does not have a corresponding point (-3, y) on the graph.**

✓ **2. Symmetry with respect to the origin:** If the graph were symmetric with respect to the origin, then for every point (x, y) on the graph, the point (-x, -y) would also be on the graph. However, the graph in the image does not appear to be symmetric with respect to the origin. For example, the point(3, y) does not have a corresponding point (-3, -y) on the graph.

✗ Since the graph does not exhibit symmetry with respect to either the y-axis or the origin, the function is neither even nor odd.**The final answer is: neither.**

Ours

✓ **Let's examine the graph of the function.** The graph shows a parabola that opens upwards and is symmetric about the y-axis. This symmetry indicates that the function is even.

✓ **To confirm, we can check a few points:**
For x = 2 , the function value is f(2). For x = -2 , the function value is f(-2) .Since the graph is symmetric about the y-axis, f(-2) = f(2) .
This confirms that f(-x) = f(x) , which is the definition of an even function.

✓ Therefore, the function is even.**The final answer is: even.**

(b) A case study from Mathverse. Base model misperceived critical visual information like symmetry and coordinates in graphs, while our model demonstrated accurate visual perception for mathematical elements.

Figure 9 | Comparison of base model and our model after rule-based RL training, showing improved visual-mathematical reasoning on geometric and coordinate problems.

We provide quantitative comparison examples below to demonstrate reasoning improvements on mathematical problems after RL training. In Fig. 9a, when solving a geometric angle problem, the base model fails to correctly interpret the critical relationship between perpendicular lines and corresponding angles. It makes contradictory assumptions about angle measures, leading to an incorrect calculation of the required rotation. In contrast, our ViGaL precisely tracks the geometric constraints and properly calculates the angle difference between initial and target positions. In Fig. 9b, when analyzing function properties from a graph, the base model incorrectly claims the function lacks symmetry despite clear visual evidence. It fails to recognize the fundamental y-axis symmetry of the parabola shown in the image. Our model immediately identifies this critical symmetrical pattern and correctly applies the appropriate mathematical definition of an even function, demonstrating enhanced visual perception of mathematical structures.