Disentangling Perception and Reasoning in Multimodal LLMs via Reward Design

Omar Sharif[♠], **Eftekhar Hossain**[♣], **Nikhil Singh**[♠*], **Patrick Ng**[†*]
[♠]Department of Computer Science, Dartmouth College
[♣]Department of Computer Science, University of Central Florida
[†] Independent Researcher
{omar.sharif.gr, nikhil.u.singh}@dartmouth.edu, eftekhar@ucf.edu

Abstract

Reinforcement learning with verifiable rewards has driven major gains in LLM reasoning, and it is intuitive to assume this recipe will transfer well to multimodal models. However, multimodal models do two things: first, perceive what is in an image, then reason about what it implies. Because these stages are graded jointly, it is hard to tell how much room reasoning alone has to grow. We study this on algorithmic visual puzzles, where both components are necessary and show that *perception*, not reasoning, is the binding constraint. Replacing images with simple textual descriptions raises performance by over 20 points on average for Claude models. We then evaluate six reward designs aimed at inducing visual grounding during reasoning without chain-of-thought supervision. Training Qwen-2.5-VL-7B with GRPO, reward design induces long, structured reasoning with self-reflection and visual references, yielding a 5.56-point gain over the base model. These gains are, however, uneven; no single reward improves all categories, and rewards with verifiable accuracy signals trade cross-domain transfer for in-domain accuracy. These results point to perception-aware reward design as a path forward, so that signals correct perception at its source rather than the reasoning that inherits its errors.

1 Introduction

We tend to evaluate multimodal models in a simplified way: the answer is right or it's not, and what happened between seeing the problem and writing the answer is effectively a black box. This obscures the fact that there are two steps in multimodal problems: perception and reasoning. For example, given an image of an analog clock and asked what time it is an hour later, the model must first see what is in the image (read the clock), and then reason about it (add an hour). A wrong answer can mean it misread the image, miscomputed from a correct reading, or both. From the answer alone, we cannot immediately tell which one is true. Indeed, while modern such models appear to excel on tasks requiring extensive domain knowledge, they often struggle with tasks that demand visual perception (Shojaee et al., 2025).

A related observation concerns how current MLLMs (Yang et al., 2024, 2025a), despite being *trained* to reason in many cases, often produce reasoning that is weakly grounded in the visual input (Fan et al., 2025a). Indeed, in our own experiments, models frequently address the algorithmic component of a question without explicit reference to the image, or jump directly to an answer with minimal intermediate description. Two failure modes thus appear together: inadequate interpretation of the image, and a tendency to bypass visual grounding altogether in the reasoning chain.

Investigating these failure modes requires both **(i)** an *evaluation setting* in which perception and reasoning can be disentangled, and **(ii)** studying *training methods* that can shape both. For (i), we turn to **visual puzzles**, which require both accurate visual perception and algorithmic reasoning. These have become an effective way to evaluate reasoning capabilities in MLLMs independent of specialized knowledge (Chia et al., 2024; Ghosal et al., 2025), showing that such models consistently underperform relative to other reasoning-oriented tasks (Song et al., 2025). They thus provide a useful setting for asking where in the pipeline performance breaks down. In other words, we ask **what is the bottleneck of MLLMs for solving visual puzzles?** Then, to separate the contributions of perception and reasoning to this gap, we hold the reasoning task constant and vary only the perceptual demand.

For (ii), we turn to reinforcement learning motivated by its empirical success in encouraging ef-

[*] Equal advising.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 2299–2322
November 6-10, 2026 ©2026 Association for Computational Linguistics

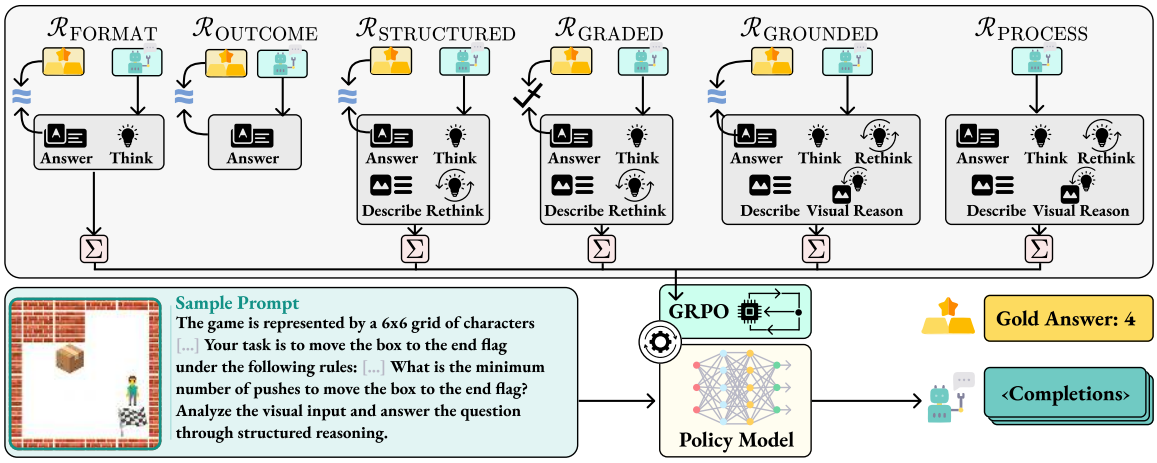


Figure 1: Overview of the GRPO-based training pipeline for MLLMs. For each sample, the policy model generates a completion conditioned on the prompt associated with a specific reward type ($\{\mathcal{R}_{\text{FORMAT}}, \mathcal{R}_{\text{OUTCOME}}, \mathcal{R}_{\text{STRUCTURED}}, \mathcal{R}_{\text{GRADED}}, \mathcal{R}_{\text{GROUNDED}}, \mathcal{R}_{\text{PROCESS}}\}$). Each completion is then evaluated using the reward calculation framework (§2). Completions with different rewards are shown in Figure 4. The resulting reward scores are passed into the GRPO trainer, which updates the policy model accordingly.

fective reasoning in text-only models. For example, generating longer textual reasoning trajectories has been shown to enable self-reflection and improve accuracy (DeepSeek-AI, 2025). The central tool for incentivizing this behavior is the *reward function* (Shao et al., 2025), and this is thus also the primary lever we use to investigate the possibilities of improving the perception–reasoning relationship. This is in contrast to recent attempts to encourage strong multimodal reasoning, which have relied instead on external supervision (Zheng et al., 2025b; Cao et al., 2025), typically in the form of annotated bounding boxes, tool outputs, or human-curated reasoning chains. **We seek a more general, supervision-free solution as have worked for text-based LLMs.** Once again, complex visual puzzles help to ground this goal in a concrete test. In these settings, such annotations are both costly to produce and difficult to define, as a single puzzle may admit multiple valid reasoning paths. The design space of multimodal reward functions is still an emerging area (Su et al., 2025).

To better understand the impact of rewards on multimodal reasoning models, **we design and evaluate six candidate reward functions** which systematically vary what information we condition on: $\{\mathcal{R}_{\text{FORMAT}}, \mathcal{R}_{\text{OUTCOME}}, \mathcal{R}_{\text{STRUCTURED}}, \mathcal{R}_{\text{GRADED}}, \mathcal{R}_{\text{GROUNDED}}, \mathcal{R}_{\text{PROCESS}}\}$ (each is defined in Table 1). These variants differ along complementary axes, such as whether the final answer must exactly match the ground truth or whether near-correct answers receive partial credit; whether intermediate visual descriptions are scored separately from the reasoning chain; and whether ground-truth supervision is provided at all. We train Qwen-2.5-VL-7B (Yang et al., 2024) with each reward under GRPO (Shao et al., 2024), and evaluate on both in-domain and cross-domain puzzle categories. Figure 1 illustrates the training pipeline.

Our results show that these post-trained models exhibit qualitatively different reasoning behavior from the base model. Under reward variants that score intermediate image descriptions, the model generates extended, structured reasoning trajectories that explicitly reference the image before performing algorithmic computation, and that revise earlier interpretations when contradictions arise. **These behaviors emerge without any chain-of-thought supervision**. The best-performing reward configuration improves accuracy on Qwen-2.5-VL-7B by 5.56 absolute points over the base model. These gains, however, do not generalize uniformly. First, reward variants that include a verifiable accuracy signal improve in-domain accuracy at a measurable cost to cross-domain transfer, suggesting overfitting to the training distribution rather than general visual reasoning improvement. Second, on categories where perceptual difficulty is highest (e.g., *Clock*), no reward variant produces substantial gains. Qualitatively, we show that the model often produces long, internally consistent reasoning trajectories that proceed from an initial misreading of the image. In other words, though the structure of the reasoning is correct, its *inputs* are not. These

Reward Function	Reward Content	Reward Calculation	Design Motivation
$\mathcal{R}_{\text{FORMAT}}$	`<think, answer>`	$\alpha_t * \text{think} + \alpha_a * \text{accuracy}$	Baseline reward for RL training for formatted reasoning and accuracy.
$\mathcal{R}_{\text{OUTCOME}}$	`<answer>`	$\alpha_a * \text{accuracy}$	Is accuracy reward enough to encourage reasoning? Can a model learn the format from the prompt alone, without any separate reward for the format?
$\mathcal{R}_{\text{STRUCTURED}}$	`<image description, think, rethink, answer>`	$\alpha_i * \text{image} + \alpha_t * \text{think} + \alpha_r * \text{rethink} + \alpha_a * \text{accuracy}$	Does distribution of reward across intuitive tags incentivize better reasoning?
$\mathcal{R}_{\text{GRADED}}$	`<image description, think, rethink, answer (partial)>`	$\alpha_i * \text{image} + \alpha_t * \text{think} + \alpha_r * \text{rethink} + \alpha_a * \text{accuracy}(partial)$	Does giving partial reward to the model when the final answer is close to the ground-truth encourage correct reasoning and answer?
$\mathcal{R}_{\text{GROUNDED}}$	`<think, visual, answer>`	$\alpha_t * \text{think} + \alpha_v * \text{visual} + \alpha_a * \text{accuracy}$	Focus on integrating visual grounding within reasoning.
$\mathcal{R}_{\text{PROCESS}}$	`<image description, think, rethink, answer-tag>`	$\alpha_i * \text{image} + \alpha_t * \text{think} + \alpha_r * \text{rethink} + \alpha_a * \text{answer-tag}$	Can the model generalize or improve its performance without any verifiable reward?

Table 1: Summary of reward functions for incentivizing visual reasoning in MLLMs. *Reward Content* specifies which output tags are evaluated, and *Reward Calculation* indicates the weighting of each reward component. Each non-answer tag's score is based on the amount of unique content generated within that tag, with counts passed through a $tanh()$ smoothing function to prevent disproportionately large rewards. Parameter values ($\alpha_t, \alpha_r, \alpha_i, \alpha_v, \alpha_a$) are provided in Appendix Table 7.

limitations suggest that reward signals applied at the level of the answer or the reasoning chain do not reliably correct errors that originate in the perceptual stage. As such, further progress in this setting will likely require reward signals that evaluate the fidelity of the model's visual interpretation directly. Overall, this work contributes:

- **Empirical evidence** that **isolates the contribution of visual perception** to multimodal puzzle-solving by substituting images with deterministic textual renderings that preserve task-relevant information. This substitution yields average gains of over 20 points for Claude models across nine puzzle categories, indicating that perception, rather than algorithmic reasoning, is the dominant constraint on performance.
- **Design and evaluation of six reward variants** that probe distinct hypotheses about how reward signals interact with visual grounding, and demonstrate that reward design alone (without chain-of-thought supervision) induces extended, image-grounded reasoning trajectories and improves Qwen-2.5-VL-7B by 5.56 absolute points.
- An **analysis of systematic limitations** of this approach, showing that reward variants with verifiable accuracy signals trade in-domain gain for cross-domain transfer, and reasoning trajectories frequently inherit rather than correct upstream perceptual errors. These results suggest that reward signals targeted at perceptual fidelity are needed for further progress.

2 Design of Reward Functions

We design six reward functions that vary along three complementary axes: (i) how the final answer is scored: by exact match, by partial credit for near-correct answers, or not at all; (ii) what intermediate output structure is rewarded: a single `<think>` block, or multiple stages spanning image description, reasoning, and self-reflection; and (iii) whether visual grounding is explicitly rewarded through a dedicated `<visual>` tag. Each variant thus probes a distinct hypothesis about which components of the reward signal contribute to long, image-grounded multimodal reasoning. Table 1 summarizes the six variants and their motivating questions; all are implemented within a single modular scoring framework, with prompts and pseudocode provided in Tables 10 and 11.

2.1 Reward Variants and Implementation

$\mathcal{R}_{\text{FORMAT}}$: Following the formulation in DeepSeek-AI (2025), this variant rewards both the presence of a `<think>` block and exact-match

accuracy on the <answer> content. It tests whether the standard text-domain recipe (i.e. format plus accuracy) is sufficient in the multimodal setting.

$\mathcal{R}_{\text{OUTCOME}}$: This variant rewards only exact-match accuracy on the <answer> content, with no credit for intermediate structure. It isolates the contribution of the accuracy signal alone and tests whether reasoning structure can be elicited from the prompt without being explicitly rewarded, motivated by DAPO (Yu et al., 2025).

$\mathcal{R}_{\text{STRUCTURED}}$: This variant distributes credit across three reasoning stages (<image description>, <think>, and <rethink>) in addition to the final answer. Each stage is scored for both multiple-tag presence and the amount of unique content within the tag, with **repetition penalized and counts smoothed by $\tanh(\cdot)$ to prevent reward hacking through redundant output**. Answer correctness is assigned the highest weight in the composite to maintain factual accuracy. The goal is to test whether explicitly rewarding multi-stage structure induces longer, more grounded reasoning trajectories without chain-of-thought supervision.

$\mathcal{R}_{\text{GRADED}}$: Identical to $\mathcal{R}_{\text{STRUCTURED}}$ but replaces exact-match accuracy with partial credit for near-correct answers (e.g. prediction of 4 or 6 when ground truth is 5). All other components (repetition penalties, $\tanh(\cdot)$ smoothing, and stage-wise structural rewards) remain unchanged, to test whether softer accuracy supervision improves reasoning, e.g. on tasks with dense numeric answer spaces.

$\mathcal{R}_{\text{GROUNDED}}$: This reward focuses on visual grounding by requiring models to integrate image-specific information into their reasoning. It introduces an explicit <visual> tag inside the <think> block and rewards unique visual descriptions within it, alongside the standard <think> and exact-match accuracy rewards. The same repetition penalty and $\tanh(\cdot)$ smoothing from $\mathcal{R}_{\text{STRUCTURED}}$ also apply. This tests whether explicitly rewarding image-referent content within the reasoning chain improves visual grounding.

$\mathcal{R}_{\text{PROCESS}}$: Finally, this configuration removes the accuracy signal entirely. The model is rewarded only for unique content in the <image description>, <think>, and <rethink> tags, and receives full credit for producing any <answer> tag regardless of its content. This effectively tests a lower bound on supervision: whether reasoning behavior can emerge from structural rewards alone, without any verifiable signal on correctness.

3 Dataset Collection and Generation

We use the AlgoPuzzleVQA dataset introduced by Ghosal et al. (2025), which comprises challenging visual puzzles designed to evaluate the capabilities of MLLMs in visual perception, language comprehension, and algorithmic reasoning. For our experiments, we selected nine categories from AlgoPuzzleVQA based on their suitability for conversion into text-only representations. This enables us to isolate and examine the reasoning bottlenecks of MLLMs without relying on visual inputs. The selected categories include: Clock, Checker Move, Maze, Move Box, N-Queens, Number Slide, Tower of Hanoi, Rotting Kiwi, and Water Jugs. Descriptions and examples are provided in Appendix C.

Training Data Generation. The dataset we use (AlgoPuzzleVQA) provides only 100 visual samples per category, which is insufficient for RL training. To scale up the dataset, we generated additional training samples by adapting the source code[1] with necessary modifications. Specifically, we created 9,000 new training samples across four categories: Clock, Maze, Move Box, and N-Queens. The Clock category contains 3,000 samples, while each of the other three categories includes 2,000 samples. These samples were generated by varying features such as the clock's shape, the color and size of the clock hands, the orientation of the grid, and the positions of the boxes and queens. The original AlgoPuzzleVQA dataset is retained as the gold test set, and we ensured that none of the generated training samples overlap with the test data. Additionally, all training examples are unique and contain no repetitions.

4 Experimental Setup

We design experiments to address three questions. First, to what extent are the limitations of MLLMs due to challenges in visual perception or algorithmic reasoning? Second, can different reward functions affect RL training to unlock long visual reasoning? Third, how well does the RL-trained model generalize to cross-distribution classes?

As our **baseline**, we evaluate the Qwen-2.5-VL-7B model using Chain-of-Thought

[1] https://tinyurl.com/4e5cvevv

prompting (<*image description*>, <*think*>, <*rethink*>, <*answer*>) without any RL fine-tuning. For the **reward-function analysis**, we apply GRPO directly to the base model without supervised fine-tuning (SFT) or a cold-start stage, as no SFT data is available. We discuss the details in Section 4.2.

Finally, we extract the answer from the model's prediction and compare it with the ground truth using exact match, reporting Accuracy as the evaluation metric (Toh et al., 2025). Prior work by Ghosal et al. (2025) evaluated model performance in both multiple-choice (MCQ) and open-ended settings. In our experiments, we found that in the MCQ setting, models often guessed the correct answer from the provided options, even when their reasoning was incorrect. This limits the evaluation's ability to accurately reflect the model's true capabilities. Therefore, we evaluate all models exclusively in the open-ended setting, where no answer options are provided. We recommend that future work adopt the open-ended format to accurately assess models' performance.

4.1 Bottleneck Experiments

To analyze the reasoning bottlenecks, we evaluate `Claude-3.5-Sonnet` and `Claude-3.7-Sonnet` in `thinking` mode across the nine selected categories. We conduct the experiments in two settings: i) `Multimodal` (`<image, question>`): where the image and corresponding question are provided to the model, ii) `Text-only` (`<text-representing image, question>`): where the image is converted into a text description, which is then passed to the model along with the question. For instance, in the 'Checker-Move' category, text descriptions are simple strings where colors are encoded as characters—'R' for red, 'G' for green, and '-' for white space. Other categories follow similarly intuitive character-based representations. Examples of how images are converted into character-based textual representations are provided in Table 8. Generic prompt structure is illustrated in Figure 2.

4.2 Training Settings

We use the `Qwen-2.5-VL-7B` (instruct) model for our experiments (Yang et al., 2024). We train the model using the GRPO (DeepSeek-AI, 2025), a reinforcement learning–based optimization method, with synthetically generated training samples (Section 3). Building on this setup, we examine how reinforcement learning may influence the model's reasoning behavior by constructing two datasets, each aligned with a complementary hypothesis. First, we hypothesize that training on challenging, high-difficulty samples can help unlock the model's longer reasoning abilities. Since Clock is one of the categories where the base model struggles most, we construct a training set consisting of 3,000 samples exclusively from this category, referred to as **Clock-3k**. Second, we hypothesize that exposure to a diverse set of reasoning categories during RL training could encourage the model to make more generalizable and transferable improvements in reasoning. To test this, we create a second training set, **Diverse-8k**, comprising 8,000 samples distributed evenly across four categories: Clock, Maze, Move-Box, and N-Queens. Further details on hyperparameters are provided in Appendix B.

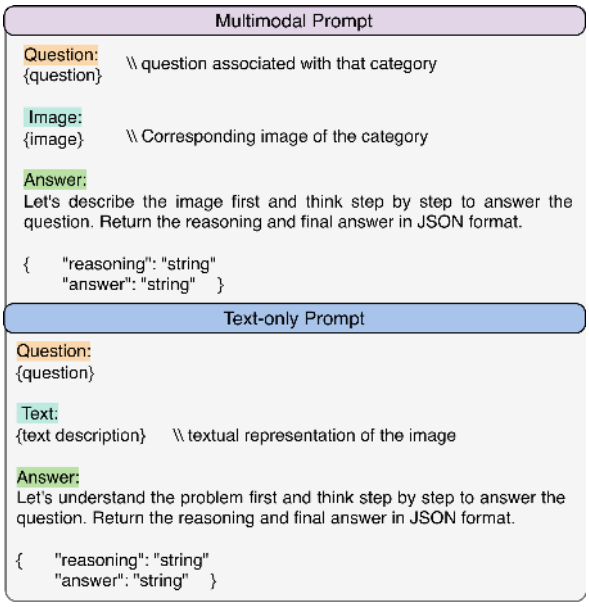


Figure 2: Generic prompt structure used across all categories in both multimodal and text-only settings.

	Claude-3.5		Claude-3.7 (thinking)	
Category	MM	TO	MM	TO
Clock	3	84	7	83
Checker move	16	9	39	44
Maze	10	15	14	26
Move box	33	58	34	70
N-queens	25	95	25	97
Number slide	36	60	74	77
Tower of Hanoi	44	60	85	85
Rotting kiwi	19	37	32	50
Water Jugs	47	55	72	62
Avg.	**25.9**	**52.6**	**42.4**	**66.0**

Table 2: Comparison of accuracy in multimodal (MM) and text-only (TO) settings in different categories.

Category		Clock-3K Data						Diverse-8k Data					
	Base	$\mathcal{R}_F$	$\mathcal{R}_O$	$\mathcal{R}_S$	$\mathcal{R}_G$	$\mathcal{R}_V$	$\mathcal{R}_P$	$\mathcal{R}_F$	$\mathcal{R}_O$	$\mathcal{R}_S$	$\mathcal{R}_G$	$\mathcal{R}_V$	$\mathcal{R}_P$
Clock	3	37	8	13	10	4	1	6	6	4	1	2	3
Checker move	6	1	3	4	5	5	10	4	10	2	10	6	1
Maze	6	0	5	10	7	0	11	26	10	12	9	30	9
Move box	25	13	24	21	17	17	19	30	43	54	30	20	21
N-queens	8	2	6	7	6	6	5	17	11	15	16	13	9
Number slide	13	4	14	16	10	10	11	7	10	10	6	10	9
Tower of hanoi	1	0	0	0	2	4	4	3	2	0	6	2	0
Rotting kiwi	24	14	22	22	18	20	28	22	30	27	27	27	29
Water jugs	3	0	1	17	16	11	13	4	14	16	27	10	20
Avg.	**10**	**7.89**	**9.22**	**12.22**	**10.11**	**8.56**	**11.33**	**13.22**	**15.11**	**15.56**	**14.67**	**13.33**	**11.22**

Table 3: Evaluation results for RL trained Qwen-2.5-VL-7B model across different reward functions. $\mathcal{R}_F$, $\mathcal{R}_O$, $\mathcal{R}_S$, $\mathcal{R}_G$, $\mathcal{R}_V$, and $\mathcal{R}_P$ denote $\{\mathcal{R}_{\text{FORMAT}}, \mathcal{R}_{\text{OUTCOME}}, \mathcal{R}_{\text{STRUCTURED}}, \mathcal{R}_{\text{GRADED}}, \mathcal{R}_{\text{GROUNDED}}, \mathcal{R}_{\text{PROCESS}}\}$ respectively. The base score indicate the performance with Chain-of-Thought approach. For each row, the best value is highlighted in a dark shade and the second-best in a light shade. All tied values within a class are highlighted in a light shade.

5 Results

5.1 Perception as primary bottleneck

Table 2 reports the performance (accuracy) of `Claude-3.5` and `Claude-3.7` across the nine puzzle categories in both the multimodal (MM) and text-only (TO) settings. Substituting the image with its textual rendering yields large and consistent gains: on average 26.7 points for `Claude-3.5` and 23.6 for `Claude-3.7`, The effect is most pronounced on categories with high perceptual demand. On *Clock*, `Claude-3.7` improves from 7% to 83%; on *N-Queens*, both models move to near-perfect accuracy. Since the question and the underlying algorithm are identical across the two conditions, the only factor that has changed is the difficulty of extracting the relevant information from the visual input. This empirically supports the hypothesis that visual perception, rather than algorithmic reasoning, is the dominant constraint.

Qualitative Analysis. To corroborate this, we inspected the multimodal-condition trajectories qualitatively, and found further support for this interpretation. Errors most often originate at the perception stage and then propagate through reasoning chains that are otherwise internally valid. For example, in the *Clock* category, models often fail to read the time from the positions of the clock hands; in *N-Queens*, they struggle to correctly identify the queens' positions on the board; and in *Move Box*, they fail to recognize the spatial positions of the person, box, and target flags. Table 9 presents a few representative examples where failures in visual reasoning lead to incorrect model predictions. Together, these patterns indicate that the reasoning components of these models are largely intact for the algorithmic tasks studied here; the bottleneck is in feeding them correct inputs.

5.2 Diverse training data yields broad gains

As illustrated in Table 3, training with the diverse dataset yields improvements across all reward functions compared to training solely on hard samples (Clock-3k). The $\mathcal{R}_{\text{STRUCTURED}}$ reward achieves the highest average score (15.56), followed closely by the $\mathcal{R}_{\text{OUTCOME}}$ reward (15.11). Interestingly, the $\mathcal{R}_{\text{PROCESS}}$ reward produces unexpected behavior: in both settings, it surpasses the baseline, and in the Clock-3k setting, it outperforms several other reward functions. These findings are consistent with Shao et al. (2025), which reports that RL training without a verifiable reward signal can still improve performance. Nevertheless, certain categories remain challenging: the model consistently struggles with Clock and Tower of Hanoi tasks, in line with Fu et al. (2025), indicating that these puzzles are intrinsically difficult due to the demands of abstract temporal and recursive reasoning. Appendix A presents additional analysis.

5.3 Structured rewards elicit visually grounded reasoning, but long reasoning alone does not guarantee accuracy

Our reward function design incentivizes the emergence of long visual reasoning in the model. Under $\mathcal{R}_{\text{FORMAT}}$ and $\mathcal{R}_{\text{PROCESS}}$ rewards, the model produces minimal intermediate reasoning, often jumping directly to the answer with little explanation of the image. Both $\mathcal{R}_{\text{STRUCTURED}}$ and $\mathcal{R}_{\text{GRADED}}$ achieve the intended behavior: as shown in Figure 4, the model generates structured reasoning steps that reference and explain different parts of

the image, exhibiting the 'think with image' phenomenon (Zheng et al., 2025b; Fan et al., 2025a). This behavior emerges across both in-domain and cross-domain categories, indicating that the reward design and training strategy successfully unlock MLLMs' long reasoning capability. However, $\mathcal{R}_{\text{OUTCOME}}$'s accuracy is on par with or better than $\mathcal{R}_{\text{STRUCTURED}}$ in about half of the classes, especially in croos-domain settings, suggesting that long reasoning alone does not guarantee improved accuracy for MLLMs (Shao et al., 2025).

5.4 An in-domain/cross-domain trade-off

To evaluate generalization, we measure performance on both in-domain (seen) and cross-domain (unseen) classes. As shown in Table 4, training on a single-category dataset (Clock-3k) results in limited generalization: although $\mathcal{R}_{\text{FORMAT}}$ achieves a large in-domain gain on the trained category (3 $\rightarrow$ 37 on *Clock*), its cross-domain accuracy drops well below the no-RL baseline (10.75 $\rightarrow$ 4.25), and 3/6 variants in total fall below the baseline on cross-domain. In contrast, the Diverse-8k dataset substantially improves performance across nearly all classes, outperforming the baselines in both in-domain (ID) and cross-domain (CD) settings, with 5/6 variants improving CD relative to baseline. The pattern across both datasets reveals a trade-off: variants that achieve the largest in-domain gains tend to generalize less well, while variants with weaker accuracy signals ($\mathcal{R}_{\text{PROCESS}}$, $\mathcal{R}_{\text{GRADED}}$) show stronger or more balanced CD performance. This is consistent with overfitting to the training distribution during RL training, an effect mitigated but not eliminated by training on a diverse set of categories. **Notably, the model demonstrates emergent reasoning capabilities across both in-domain and cross-domain categories (Figures 14 and 15).**

	Clock-3K		Diverse-8k	
	ID	CD	ID	CD
Baseline	3	10.75	10.5	9.4
$\mathcal{R}_{\text{FORMAT}}$	37	4.25	19.75	8
$\mathcal{R}_{\text{OUTCOME}}$	8	9.37	17.5	13.2
$\mathcal{R}_{\text{STRUCTURED}}$	13	12.12	21.25	11
$\mathcal{R}_{\text{GRADED}}$	10	10.12	14.00	15.20
$\mathcal{R}_{\text{GROUNDED}}$	4	9.12	16.25	11
$\mathcal{R}_{\text{PROCESS}}$	1	12.62	10.50	11.80

Table 4: In-domain (ID) and cross-domain (CD) performance of the RL-trained Qwen-2.5-VL-7B model under different reward functions. For Clock-3K, the Clock category is treated as in-domain, while the remaining eight categories are considered cross-domain. For Diverse-8K, the aggregate performance over four categories—Clock, Maze, Move-Box, and N-Queens is reported as in-domain, while the remaining five categories are considered cross-domain.

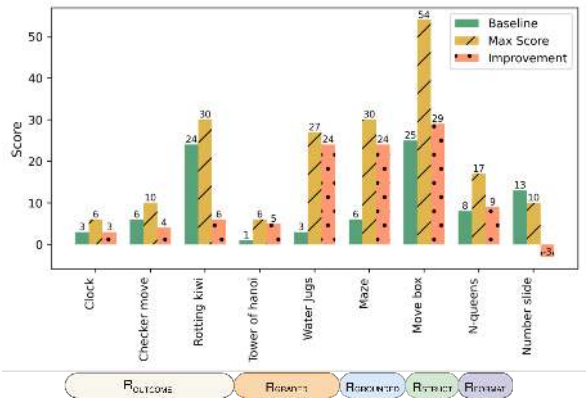


Figure 3: Performance comparison across categories showing baseline vs. maximum scores and improvement margins. Reward functions that achieved the maximum score are indicated below each category. For the number slide category no reward improve over the baseline. Results are reported for models trained on the Diverse-8K dataset.

To further investigate generalization, we explore the best-performing reward function for each category with the model trained on the Diverse-8K dataset. Detailed class-specific results are provided in Figure 3. No single reward function consistently dominates across all classes; some reward functions excel in certain categories but underperformed in others. Five different reward functions achieved the maximum score in different categories. For the number slide class, performance dropped with RL training. This variability highlights the heterogeneity in task difficulty and model behavior. The variability in reward function performance arises because puzzle categories differ in their reasoning demands. We group them into two types:

- **Visual dependency** (e.g., Move Box, Maze): These puzzles require correctly reading spatial layouts, object positions, and grid configurations from the image. Rewards such as $\mathcal{R}_{\text{STRUCTURED}}$ or $\mathcal{R}_{\text{GROUNDED}}$, which encourage visual reasoning, are particularly effective for these tasks because they push the model to attend to relevant image details before reasoning. Table 5 shows the effectiveness on reasoning oriented rewards compare to outcome-only reward and baseline.

- **Algorithmic dependency**: RL training gener-

ally improves performance over the baseline across these categories (Table 3). However, these puzzles (e.g., Checker Move, Tower of Hanoi) have relatively simple visual inputs but require more complex recursive or temporal reasoning. In such cases, encouraging additional visual reasoning may provide limited benefit as they might distract the model from the algorithmic steps needed to solve the task.

Appendix A presents additional analysis of how rewards affect performance.

Category	CoT	$\mathcal{R}_O$	$\max(\mathcal{R}_S, \mathcal{R}_G, \mathcal{R}_V)$
Maze	6	10	30
Move box	25	43	54
Rotting kiwi	24	30	27
Water jugs	3	14	27

Table 5: Head-to-head comparison of the outcome-only reward ($\mathcal{R}_{\text{OUTCOME}}$) and reasoning-oriented rewards ($\mathcal{R}_{\text{STRUCTURED}}$, $\mathcal{R}_{\text{GRADED}}$, $\mathcal{R}_{\text{GROUNDED}}$) on visually dependent categories. The best reasoning-oriented reward improves performance by up to 29 points over the CoT baseline and consistently outperforms the outcome-only reward illustrating the impact of longer reasoning.

5.5 Error Analysis

Despite being our best-performing model, the RL-trained Qwen-7B achieves only 15.56% accuracy. To understand this limited performance, we qualitatively analyze the errors and identify four primary causes of failure. First, the model makes visual understanding errors even after generating extensive visual reasoning, it incorrectly identifies object positions or attributes in the input image. See the generated completion with the $\mathcal{R}_{\text{STRUCTURED}}$ reward in Figure 4. Second, the model also fails at algorithmic reasoning, generating erroneous intermediate computations or decision steps.

Miscalculation of time

....<intermediate reasoning>....
Subtract the Time: Start with the hours. Subtracting 2 hours from 1 o'clock results in 9 o'clock.
....<continue reasoning>....

Third, we observe inconsistencies in which the model's final answer contradicts its prior reasoning.

Contradiction with own reasoning

....<intermediate reasoning>....
Therefore, the time is confidently determined to be **12:35**.
<answer> 12.37 <answer>

Finally, for a small number of cases, the model fails to produce a valid answer, continuing to generate tokens without reaching a definitive conclusion.

Continue generation without answer

....<intermediate reasoning>....
The hour hand is closest to the 1.
The hour hand is not aligned with the 11
....<continue reasoning>....
Do not generate any <answer> tag.

Overall, the error analysis highlights the multifaceted challenges posed by visual puzzle tasks. Our findings suggest that, beyond encouraging extended reasoning, this task may require reward signals that explicitly incentivize accurate visual grounding. For example, evaluating the intermediate reasoning with a strong LLM-based judge. Incorporating such signals represents a promising direction for future research.

6 Related Work

Research on the visual reasoning abilities of MLLMs has grown rapidly (Chen et al., 2024; Li et al., 2025b; Zhang et al., 2024), with benchmarks spanning abstract visual puzzles (Song et al., 2025), diagrammatic and symbolic reasoning (Li et al., 2025a), and algorithmic visual reasoning (Ghosal et al., 2025). Across these settings, MLLMs consistently struggle to uncover the visual relations necessary for correct inference. While CoT prompting elicits reasoning effectively in text-only domains (Wei et al., 2022; Kojima et al., 2022), its visual extensions often produce interpretations weakly grounded in the image (Dong et al., 2025; Zhang et al., 2024). Because existing studies evaluate on prediction accuracy alone, they offer limited insight into whether errors stem from visual misinterpretation or faulty algorithmic reasoning (Ghosal et al., 2025; Bi et al., 2025). We instead perform a systematic inspection to isolate the source of error.

Reinforcement Learning has recently emerged as a way to improve reasoning in MLLMs. This is primarily motivated by the advances in process supervision and verifiable reward modeling from text-only systems (Lightman et al., 2023; DeepSeek-AI, 2025). In the visual setting, these

ideas are adapted by using reward signals that verify whether intermediate reasoning steps are consistent with the image. For instance, vision-language process reward models (Wang et al., 2025; Ong et al., 2025) provide feedback on reasoning steps by checking them against visual evidence. In parallel, some studies explore multimodal extensions of R1-style training to elicit longer reasoning trajectories for diagrams, mathematical problems, and synthetic visual tasks (Yang et al., 2025b; Shen et al., 2025; Zhou et al., 2025). Complementary approaches such as GRIT (Fan et al., 2025b) and DeepEyes (Zheng et al., 2025c) incorporate explicit visual grounding, encouraging models to reference image regions during multi-step explanations. Liang et al. (2025) optimize the data mixture with verifiable rewards to improve reasoning.

However, these approaches rely on intermediate supervision such as executable tool outputs, bounding boxes, or annotated reasoning traces. An open question thus remains as to whether structured reasoning can be induced through reward design. We take a step toward addressing this by designing reward functions and systematically evaluating their impact on encouraging long, structured visual reasoning without requiring additional supervision.

7 Conclusion

Our results suggest a pressing need to decouple perception and reasoning in post-training for multimodal models. Today's conventional recipes, largely adapted from text, apply reward at the level of the answer or the reasoning chain and leave the perceptual stage to improve as a side effect. We argue this is not sufficient: long, structured reasoning can faithfully propagate whatever the model *thought* it saw. The six reward variants we study begin to establish how supervision can be distributed across these stages and isolate which signals shape which behaviors. We see them as a first step toward a broader program in which perception and reasoning are trained, rewarded, and evaluated as the distinct competencies they are.

Limitations

Our primary objective is to investigate whether reward-based RL fine-tuning can encourage models to perform long, multi-step visual reasoning required to solve complex puzzles. While alternative approaches such as SFT could potentially boost performance on a specific dataset, they cannot guide a model to exhibit step-by-step reasoning behavior. Another possibility could have been SFT with explicit reasoning traces. However, for visual puzzle datasets, collecting high-quality chain-of-thought annotations is extremely challenging and costly, mainly because the reasoning is often multi-hop, brittle, and domain-specific. For these reasons, we focus on reward-driven RL fine-tuning, where the aim was not merely to fit the dataset but to induce emergent long-horizon reasoning behavior, which we believe is better aligned with the nature of visual puzzle solving.

Another concern is that the textual representations of images are highly structured and likely overstate the perception gap. However, the purpose of this experiment is to establish an upper bound that quantifies the maximum performance gap attributable to visual perception. The text representations are kept minimal (e.g., "G-RR" for checker states, simple grid characters); they encode the same spatial information present in the image without introducing any reasoning shortcuts. Moreover, there is no perfect middle ground between an image and structured text for isolating the visual perception bottleneck from reasoning capability. Possible alternatives, such as noisy or partially corrupted text descriptions or caption-based representations generated by a vision model, each introduce their own biases and assumptions, making it difficult to draw clean conclusions. Our approach instead provides a clear and interpretable upper bound that cleanly separates the perception bottleneck from reasoning capability.

References

Jing Bi, Guangyu Sun, Ali Vosoughi, Chen Chen, and Chenliang Xu. 2025. Diagnosing visual reasoning: Challenges, insights, and a path forward. *arXiv preprint arXiv:2510.20696*.

Meng Cao, Haoze Zhao, Can Zhang, Xiaojun Chang, Ian Reid, and Xiaodan Liang. 2025. Ground-r1: Incentivizing grounded visual reasoning via reinforcement learning. *Preprint*, arXiv:2505.20272.

Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. 2024. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1254–1262.

Yew Ken Chia, Vernon Toh, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. 2024. PuzzleVQA: Diagnosing multimodal reasoning challenges of language

models with abstract visual patterns. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16259–16273, Bangkok, Thailand. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. 2025. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9062–9072.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. 2025a. Grit: Teaching mllms to think with images. *Preprint*, arXiv:2505.15879.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. 2025b. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*.

Tairan Fu, Miguel González, Javier Conde, Elena Merino-Gómez, and Pedro Reviriego. 2025. Have multimodal large language models (mllms) really learned to tell the time on analog clocks? *Preprint*, arXiv:2505.10862.

Deepanway Ghosal, Vernon Toh, Yew Ken Chia, and Soujanya Poria. 2025. AlgoPuzzleVQA: Diagnosing multimodal reasoning challenges of language models with algorithmic multimodal puzzles. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9615–9632, Albuquerque, New Mexico. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Can Li, Ying Liu, Ting Zhang, Mei Wang, and Hua Huang. 2025a. Visiomath: Benchmarking figure-based mathematical reasoning in lmms. *arXiv preprint arXiv:2506.06727*.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuan-Jing Huang, and Zhongyu Wei. 2025b. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3769–3798.

Yiqing Liang, Jielin Qiu, Wenhao Ding, Zuxin Liu, James Tompkin, Mengdi Xu, Mengzhou Xia, Zhengzhong Tu, Laixi Shi, and Jiacheng Zhu. 2025. Modomodo: Multi-domain data mixtures for multimodal llm reinforcement learning. *Preprint*, arXiv:2505.24871.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Brandon Ong, Tej Deep Pala, Vernon Toh, William Chandra Tjhi, and Soujanya Poria. 2025. Training vision-language process reward models for test-time scaling in multimodal reasoning: Key insights and lessons learned. *arXiv preprint arXiv:2509.23250*.

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. 2025. Spurious rewards: Rethinking training signals in rlvr. *Preprint*, arXiv:2506.10947.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, and 1 others. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *Preprint*, arXiv:2506.06941.

Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. 2025. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *Preprint*, arXiv:2504.10342.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, Linjie Li, Yu Cheng, Heng Ji, Junxian He, and Yi R. Fung. 2025. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *Preprint*, arXiv:2506.23918.

Vernon Y. H. Toh, Yew Ken Chia, Deepanway Ghosal, and Soujanya Poria. 2025. The jumping reasoning curve? tracking the evolution of reasoning performance in gpt-[n] and o-[n] models on multimodal puzzles. *Preprint*, arXiv:2502.01081.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, and 1 others. 2025. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, and et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, and 1 others. 2025b. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yizhe Zhang, He Bai, Ruixiang Zhang, Jiatao Gu, Shuangfei Zhai, Josh Susskind, and Navdeep Jaitly. 2024. How far are we from intelligent visual deductive reasoning? *arXiv preprint arXiv:2403.04732*.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. 2025a. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025b. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. *Preprint*, arXiv:2505.14362.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025c. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*.

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

Appendix

A Additional Result Analysis

We perform additional analysis of how reward design affects performance. Tables 3 and 4 surface several patterns worth examining: $\mathcal{R}_{\text{OUTCOME}}$, which uses no reasoning signal, is often competitive with more complex rewards, and $\mathcal{R}_{\text{PROCESS}}$, which does not grade the answer, achieves the strongest cross-domain generalization. We account for these as follows.

- **Why $\mathcal{R}_{\text{OUTCOME}}$ is competitive.** This reward leaves the model free to reason however it wants, which suffices for some tasks and yields competitive accuracy. The difference is qualitative rather than quantitative: $\mathcal{R}_{\text{OUTCOME}}$ does not produce the extended reasoning that $\mathcal{R}_{\text{STRUCTURED}}$ does (Figure 4). Accuracy is close, but reasoning quality differs substantially, and that quality is what we aim to improve.

- **Why $\mathcal{R}_{\text{PROCESS}}$ generalizes in cross-domain.** A strong accuracy signal pushes the RL-trained model toward task-specific answers, raising in-domain accuracy at the cost of transfer. $\mathcal{R}_{\text{PROCESS}}$ sidesteps this by design: it rewards only general reasoning behavior (describing the image, thinking, rethinking), which carries over to unseen tasks.

- **Why some rewards show large ID–CD gaps.** Rewards with high accuracy weight ($\alpha_a = 0.80$–0.90) reinforce task-specific patterns and widen the gap. Rewards that spread weight across reasoning components ($\mathcal{R}_{\text{PROCESS}}$) or relax accuracy matching ($\mathcal{R}_{\text{GRADED}}$) reward more transferable skills, producing more balanced performance

B Implementation Details

We experimented with two pre-trained MLLMs `Qwen-2.5-VL-3B` and `Qwen-2.5-VL-7B`. The 3B parameter model exhibited poor performance in our preliminary experiments and was therefore excluded from further evaluation. We train the 7B model using the GRPO (DeepSeek-AI, 2025) approach with a learning rate of 1e-6 and a batch size of 128. Parameters for training are presented in Table 6. To prevent memory overflow, we set both the maximum prompt length and response length

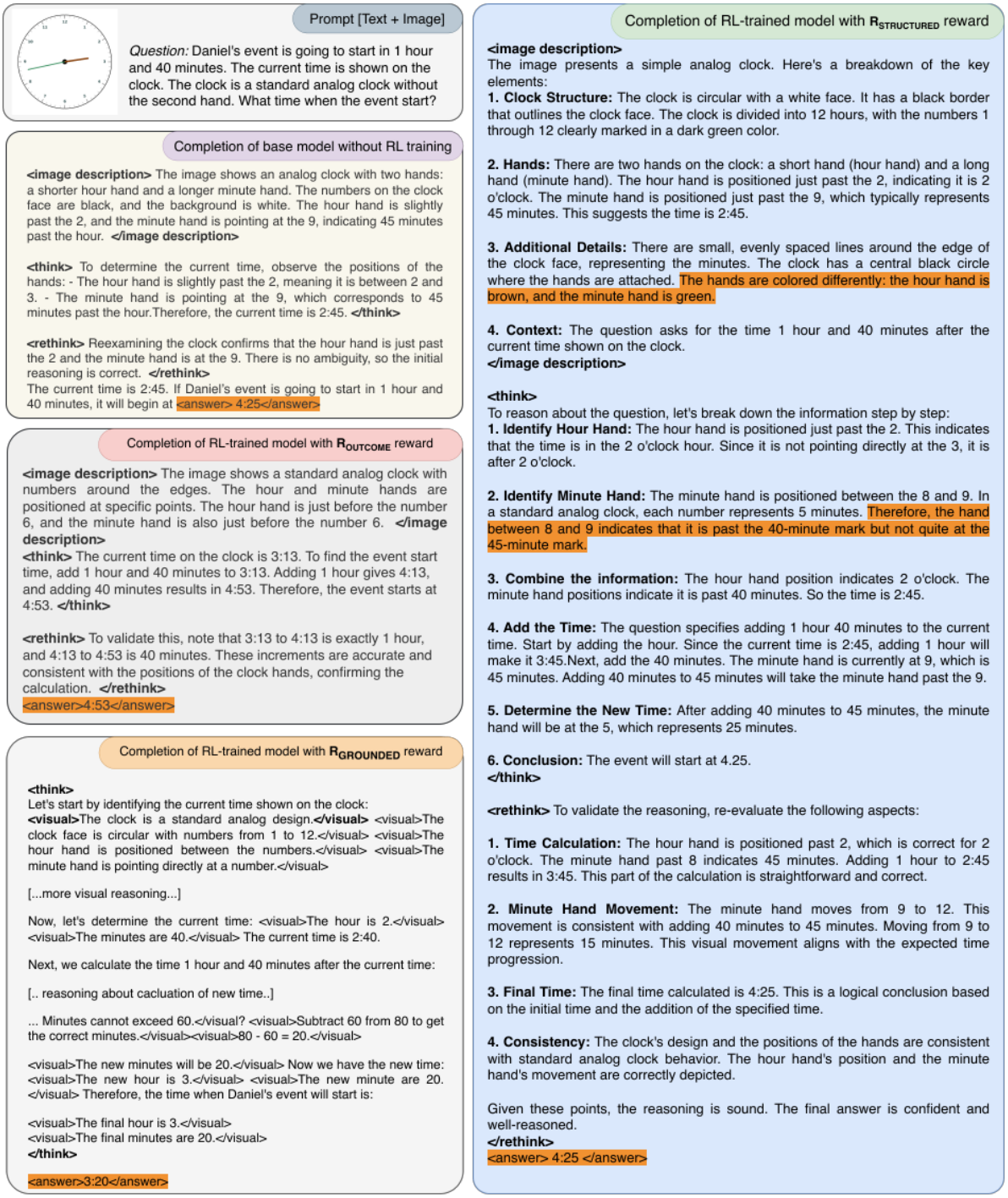


Figure 4: An example comparing reasoning between the base (Qwen-2.5-VL-7B) with chain-of-thought approach and RL-trained versions with different reward strategies. With the $\mathcal{R}_{\text{STRUCTURED}}$ reward, the model generates grounded, multi-step visual reasoning–accurately describing fine-grained image details such as the distinct colors of the hour and minute hands. Notably, this structured long-form reasoning emerges naturally from RL training (see Figures 14 and 15 for additional examples). The model also performs step-by-step self-evaluation, revisiting and verifying each reasoning step before concluding with its final answer.

to 2,048 tokens and train the model for 500 steps. The best intermediate model is saved based on the validation score. The KL divergence coefficient is set to 0.0 to promote diverse reasoning strategies. All experiments are conducted on NVIDIA H100 GPUs. We use the EasyR1 (Zheng et al., 2025a) framework to implement our RL models.

Setting	Value
Training Steps	500
Batch Size	128
Max Prompt Length	2048
Max Response Length	2048
Max Image Pixels	262144
Min Image Pixels	65536
Learning Rate	$1e^{-6}$
Temperature	1.0
KL Coefficient	0.00

Table 6: Parameters for the GRPO training

Reward Function	Parameter Weights
$\mathcal{R}_{\text{FORMAT}}$	$\alpha_t = 0.10$, $\alpha_a = 0.90$
$\mathcal{R}_{\text{OUTCOME}}$	$\alpha_a = 1.0$
$\mathcal{R}_{\text{STRUCTURED}}$, $\mathcal{R}_{\text{GRADED}}$	$\alpha_i = 0.06$, $\alpha_t = 0.06$, $\alpha_r = 0.08$, $\alpha_a = 0.80$
$\mathcal{R}_{\text{GROUNDED}}$	$\alpha_t = 0.10$, $\alpha_v = 0.10$, $\alpha_a = 0.80$
$\mathcal{R}_{\text{PROCESS}}$	$\alpha_i = 0.30$, $\alpha_t = 0.30$, $\alpha_r = 0.30$, $\alpha_a = 0.10$

Table 7: Weight configurations used across different reward functions. To ensure factual accuracy, a high weight is consistently assigned to answer correctness (α_a). The other components are weighted based on empirical observations.

C Dataset Details

A description of selected categories from AlgoPuzzleVQA is provided below.

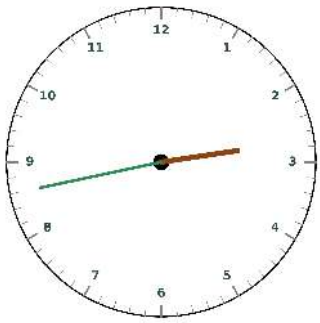


Figure 5: `Question: Emily's event is going to start in 1 hour 10 minutes. The current time is shown on the clock. The clock is a standard analog clock without the second hand. What time will the event start?`

Clock: This category features analog clocks with only hour and minute hands, displaying a randomly chosen current time. Each question describes an event that occurred (or will occur) *h* hours and *m* minutes before or after the current time, and the model must infer the correct time at which the event took place. An example is shown in Figure 5.

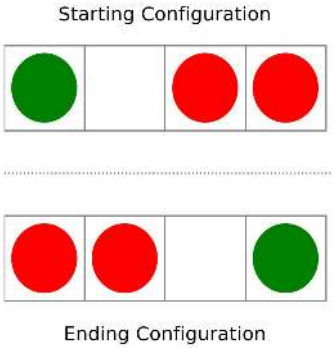


Figure 6: `Question: A checker game is being played on a grid of 4 squares with 1 green and 2 red checkers. Initially, the checkers are arranged as shown in the starting configuration, with the 3 checkers occupying 3 squares and one unoccupied square. Green checkers only move rightward and red checkers only move leftward. Every move is either i) a slide to the adjacent empty square, or ii) a jump over one position to an empty square, provided the checker being jumped over is of a different color. Each square can accommodate a maximum of one checker at any time. How many moves are required to reach the ending configuration from the starting configuration following the specified rules?`

Checker Move: This puzzle is played on a one-dimensional grid consisting of *n* cells, where *n–1* cells are initially occupied by either red or green checkers. The objective is to transform the starting configuration into a specified ending configuration, subject to the following constraints: (i) red checkers can only move leftward, and green checkers can only move rightward; (ii) each move must be either a slide into the adjacent empty cell or a jump over one checker of the opposite color into an empty cell; and (iii) each cell may hold at most one checker at any given time. The task is to compute the minimum number of moves required to reach the goal configuration from the initial state. Each puzzle is guaranteed to have at least one valid solution. A sample puzzle is illustrated in Figure 6.

Maze: This is a path-finding puzzle with predefined start and end positions. White cells indicate valid paths, while black cells represent walls. The model must answer one of two types of questions:

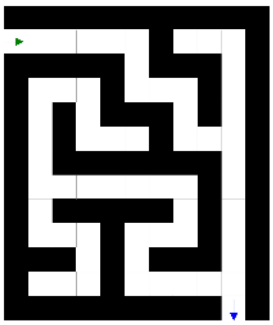


Figure 7: Question: This is maze having 13 * 11 cells. The empty cells are coloured white and the obstacle cells are coloured black. From an empty cell, you can only move up, down, left, or right to another adjacent empty cell. You cannot move diagonally between two empty cells and cannot step into a cell with an obstacle. The entry cell of the maze is shown with the green arrow. The exit cell of the maze is shown with the blue arrow. Suppose you have found the most optimal path in the maze between the entrance and exit, where you need to go through the least number of empty cells and you need to make the least number of left and right turns. What is the total number of left turns do you need to make in this optimal path?

(i) How many left, right, or total turns are required in the optimal path? or (ii) How many cells must be visited in the optimal path, including the entrance and exit? An example maze and corresponding question are shown in Figure 7.

Move Box: This puzzle involves a game with a person, a box, and a target location, all placed on a grid with obstacles. The objective is to move the box to the target location using the minimum number of pushes. This puzzle can be solved using the Breadth-First Search (BFS) technique. An example shown in Figure 8.

N-Queens: This puzzle is based on the classic chess problem where the objective is to place N queens on an $N \times N$ chessboard such that no two queens share the same row, column, or diagonal. In this variation, $N - 2$ queens are already placed on the board. The question includes what the Manhattan distance (i.e., the number of unique squares on the board) is between the two remaining queens when they are placed correctly? The solution can be achieved backtracking algorithm. Figure 9 illustrates an n-queens puzzle.



Figure 8: Question: A storekeeper is a puzzle in which the player pushes boxes around in a warehouse trying to get them to target locations. The game is represented by an 6 x 6 grid of characters grid where each element is a wall, floor, or box. Your task is to move the box to the end flag under the following rules: 1. The box can be moved to an adjacent free cell by standing next to the box and then moving in the direction of the box by 1 grid. This is a push. 2. The player cannot walk through the box. What is the minimum number of pushes to move the box to the end flag.

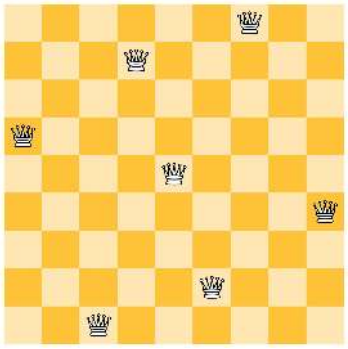


Figure 9: Question: You are given a 9 * 9 chessboard. The Manhattan distance between two squares in a chessboard is equal to the minimal number of orthogonal King moves between these squares on the otherwise empty board. The objective is to place 9 chess queens on this board so that no two queens threaten each other; i.e. no two queens share the same row, column, or diagonal. 7 queens have already been placed in some of the squares of the board, as shown in the image. Suppose you pick two squares to place the two remaining queen pieces in a way that fulfills the objective. What is the Manhattan distance between these two squares?

Number Slide: This puzzle consists of an $N * N$ board with numbered tiles from 1 to $(N * N) - 1$

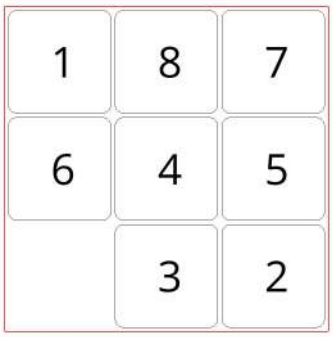


Figure 10: Question: The board shown in the image is a sliding puzzle of 3 * 3 tile dimensions. It has 8 numbered tiles and one unoccupied (open) position. Tiles in the same row or column of the open position can be moved by sliding them horizontally or vertically, respectively. All tiles always stay and move inside the red boundary wall, as shown in the image. A move is defined as moving the open position by one tile unit in any available direction. You start from the board position shown in the image and perform exactly 4 moves such that the open position is seen moving in the following sequence: up, up, right, left. What is the maximum number in the row that now has the open position?

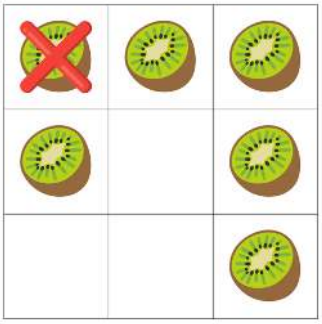


Figure 11: Question: You are given a 3 x 3 grid in which each cell can contain either no kiwi, one fresh kiwi, or one rotten kiwi. Every minute, any fresh kiwi that is 4-directionally adjacent to a rotten kiwi also becomes rotten. What is the minimum number of minutes that must elapse until no cell has a fresh kiwi?

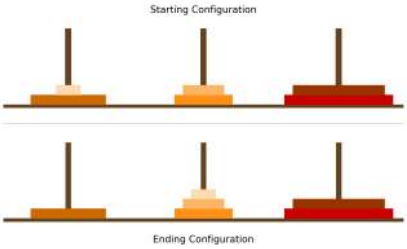


Figure 12: Question: You are playing a Tower of Hanoi game with 3 rods and 6 disks of various diameters, which can slide onto any rod. You are given the starting and ending configuration of the game as shown in the top and the bottom of the image, respectively. The game has the following rules: i) Only one disk may be moved at a time; ii) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod; and iii) No disk can be placed on top of a disk that is smaller than it. What is the minimum number of moves required to go from the starting to the ending configuration?

and one empty position. The numbers are arranged randomly, and the puzzle involves answering one of the following types of questions: i) How many unique board positions can be reached after performing n moves? ii) What is the minimum or maximum sum that can be achieved in a particular row or column after performing n moves? iii) Perform n moves where the empty position can be moved in (up, down, left, right) ways. Now, what is the minimum or maximum sum that can be achieved in a particular row or column that has the empty position? Figure 10 shows a $3 * 3$ puzzle with a sample question.

Rotting Kiwi: The puzzle involves a rectangular grid containing fresh and rotten kiwis. Initially, one kiwi is rotten. As time progresses, the rot spreads to adjacent kiwis (up, down, left, right) each minute. The objective is to determine the earliest time at which all kiwis become rotten. An example puzzle is illustrated in Figure 11.

Tower of Hanoi: This puzzle involves solving a mathematical game consisting of 3 rods and n disks of varying diameters. You are given an initial and a target configuration of the towers. The objective is to reach the target configuration using the minimum number of moves. The game follows these rules: i) Only one disk can be moved at a time, ii) Each move involves taking the top disk from one stack and placing it on top of another stack or an empty rod, iii) No disk may be placed on top of a smaller disk. It is guaranteed that an optimal solution exists. An example puzzle is shown in Figure 12.

Water Jugs: This is a measurement-based puzzle involving n jugs with integer capacities. The initial amount of water in each jug is provided as visual

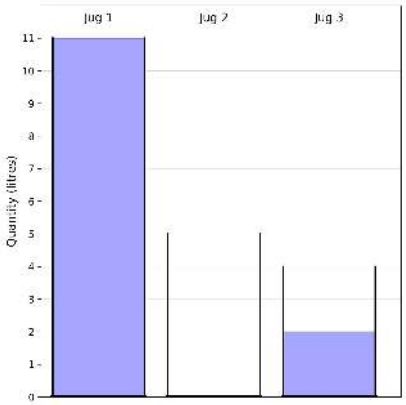
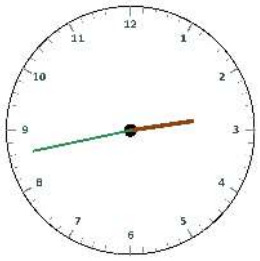
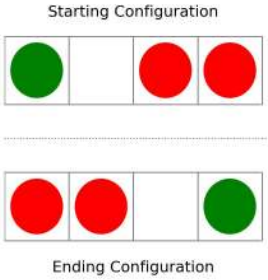
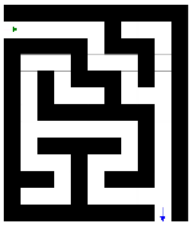
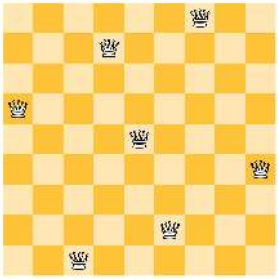
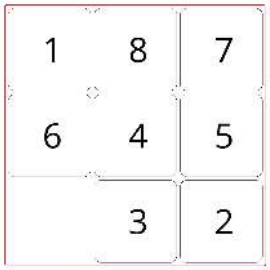


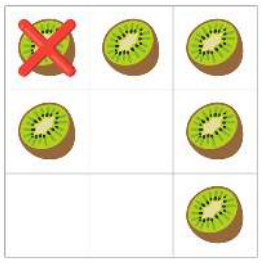
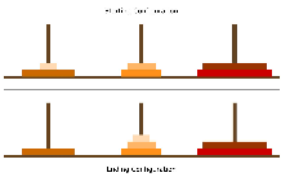
Figure 13: Question: You are given 3 jugs of capacities 11, 5, 4 litres. Initially, the amount of water that is contained in each jar is shown in the image. A single step of water pouring from one jug to another is constrained by the following rules: i) take a non-empty jug and pour water from it to another non-full jug until the first one becomes empty or the second one becomes full; and ii) no water can be split while pouring. The objective is to reach the amounts of 11, 1, 1 litres of water in the jugs from left to right, respectively. What is the minimum number of water pouring steps required to achieve the objective?

context. The model is then asked to determine the number of pouring steps required to reach a goal state, which specifies the desired quantity of water in each jug. Water pouring is subject to the following constraints: i) Water can be poured from a non-empty jug into another non-full jug until either the source jug is empty or the target jug is full, ii) No water can be spilled during the process. Both the initial and goal states are randomly generated with a solution. An example is shown in Figure 13.

Image	Textual Representation	Image	Textual Representation
	`Current Time: 2:43`		`Starting Configuration: G-RR` `Ending Configuration: RR-G`

	The maze is illustrated in the following grid, where white empty cells are denoted by `0` and black obstacle cells are denoted by `1`.
	`1 1 1 1 1 1 1 1 1 1 1`
	`0 0 0 0 0 0 1 0 0 0 1`
	`1 1 1 1 1 0 1 1 1 0 1`
	`1 0 0 0 1 0 0 0 1 0 1`
	`1 0 1 0 1 1 1 0 1 0 1`
	`1 0 1 0 0 0 1 0 0 0 1`
	`1 0 1 1 1 1 1 1 1 0 1`
	`1 0 0 0 0 0 0 0 1 0 1`
	`1 0 1 1 1 1 1 0 1 0 1`
	`1 0 0 0 1 0 0 0 1 0 1`
	`1 1 1 0 1 0 1 1 1 0 1`
	`1 0 0 0 1 0 0 0 0 0 1`
	`1 1 1 1 1 1 1 1 1 0 1`
			Grid is shown below, where the wall, floor, and box are represented by characters `#`, `.`, and `B` respectively. The person is denoted by `S` and the end flag is by `T`.
			`######`
			`###..#`
			`#.B..#`
			`#...S#`
			`#...T#`
			`######`

	Queen positions in the chessboard are given in the following lists.
	`[0, 6]`
	`[1, 3]`
	`[3, 0]`
	`[4, 4]`
	`[5, 8]`
	`[7, 5]`
	`[8, 2]`
			The image/board is represented as a grid, and the unoccupied (open) position is denoted with `0`.
			`1 8 7`
			`6 4 5`
			`0 3 2`

	Grid is shown below, where `1` denotes fresh kiwis, `2` denotes rotten kiwis, and `0` represents empty spaces.
	`2 1 1`
	`1 0 1`
	`0 0 1`
			The starting and ending configurations of the towers are given in the following lists.
			`[4, 1], [3, 2], [6, 5]`
			`[4], [3, 2, 1], [6, 5]`

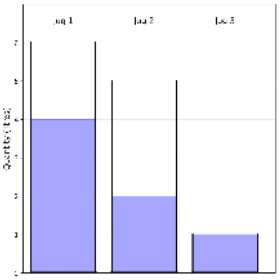
	The capcities, initial amount of water in each jar and objective amount of water in each jar are given in the following lists. Capacities: `[6, 5, 1]` Initial: `[4, 2, 1]` Objective: `[4, 3, 0]`		

Table 8: Examples of how images from different classes are converted into textual representations. In the multimodal setting, images are provided to the model, while in the text-only setting, their textual descriptions are used.

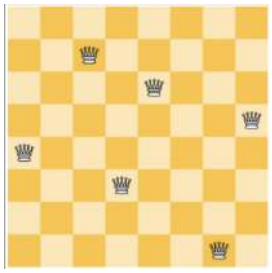
Image	Question	Reasoning, Answer, and Comment
	Question: Daniel's event is going to start in 1 hour 40 minutes. The current time is shown on the clock. The clock is a standard analog clock without the second hand. What will be the time when the event starts? *Ground-Truth:* `4:23`	***Reasoning***: The clock shows an analog face with hour and minute hands. The minute hand (brown) is pointing to 3, which indicates 15 minutes past the hour. The hour hand (green) is between 8 and 9, but closer to 8, confirming it's 8:15. To find when the event starts, I need to add 1 hour and 40 minutes to 8:15. Adding 1 hour gives 9:15, and adding 40 more minutes results in 9:55. ***Answer***: 9:55 ***Comment***: `Incorrectly interpret the positions of the minute and hour hands, leading to interpreting the wrong time. But the calculation of time is correct.`
	Question: A storekeeper is a puzzle in which the player pushes boxes around in a warehouse trying to get them to target locations. The game is represented by an 6 x 6 grid of characters grid where each element is a wall, floor, or box. Your task is to move the box to the end flag under the following rules: 1. The box can be moved to an adjacent free cell by standing next to the box and then moving in the direction of the box by 1 grid. This is a push. 2. The player cannot walk through the box. What is the minimum number of pushes to move the box to the end flag. *Ground-Truth:* `4`	***Reasoning***: The puzzle shows a maze with brick walls, a box in the center, and a checkered flag on the right. The player starts at the bottom center. To solve it with the fewest pushes, the player first moves up to reach the box. Then, based on the layout, the box should be pushed to the right toward the flag. The box is about two grid spaces away, with no obstacles requiring extra moves. Therefore, the minimum number of pushes needed is 2. ***Answer***: 2 ***Comment***: `Incorrect recognition of the spatial positions in the grid, leading to misidentification of the positions of obstacles, boxes, and the flag.`
	Question: You are given a 8 * 8 chessboard. The Manhattan distance between two squares in a chessboard is equal to the minimal number of orthogonal King moves between these squares on the otherwise empty board. The objective is to place 8 chess queens on this board so that no two queens threaten each other; i.e. no two queens share the same row, column, or diagonal. 6 queens have already been placed in some of the squares of the board, as shown in the image. Suppose you pick two squares to place the two remaining queen pieces in a way that fulfills the objective. What is the Manhattan distance between these two squares? *Ground-Truth:* `10`	***Reasoning***: Looking at the 8×8 chessboard, I see six queens already placed. I'll use coordinates where (1, 1) is the top-left and (8, 8) is the bottom-right. The six queens are at: - (2, 3) – 2nd row, 3rd column - (3, 6) – 3rd row, 6th column - (4, 8) – 4th row, 8th column - (5, 1) – 5th row, 1st column - (6, 4) – 6th row, 4th column - (8, 7) – 8th row, 7th column To place the remaining two queens, I must find squares that are not in the same row, column, or diagonal as any existing queen. After marking all threatened squares, only two valid positions remain: (1, 5) and (7, 2). The Manhattan distance between them is: |1-7|+|5-2|=6+3=9 ***Answer***: 9 ***Comment***: `The queens' positions on the board were misidentified, which resulted in a wrong answer. The subsequent reasoning and calculations are accurate.`

Table 9: Qualitative example illustrating failure in visual perception leading to incorrect model predictions. Incorrect visual generations are highlighted in red, with our comments on the visual errors provided.

Long and grounded visual reasoning of RL trained Qwen-2.5-7B model with $R_{STRUCTURED}$ Reward



Move box puzzle

<image description>
The image presents a simple puzzle setup. Here's a breakdown of the visual elements:

1. Background: The entire image is framed by a brick wall. The bricks are uniformly red and arranged in a grid pattern, creating a clear boundary for the puzzle area.

2. Box: There is a single cardboard box in the center of the image. The box is brown with a simple design, indicating it is the object to be moved. It is positioned in the middle of the grid.

3. Player: To the right of the box, there is a small figure representing a person. This figure is yellow with a simple design, suggesting it is the player. The player is positioned to the right of the box.

4. Flag: Below the player, there is a checkered flag. The flag is black and white, with a classic racing flag design. It is positioned on the floor, indicating the target location for the box.

5. Grid Layout: The entire area is divided into a 6x6 grid. The grid lines are faint, but they help in visualizing the movement of the box. The box is currently in the center of the grid, and the flag is in the bottom-right corner.

6. Movement Constraints: The box can only be pushed to an adjacent free cell. The player cannot walk through the box, which means the player must stand next to the box to push it.

The puzzle requires the player to move the box from its current position to the flag. The task is to determine the minimum number of pushes needed to achieve this.
</image description>

<think>
To solve this puzzle, we need to reason through the steps required to move the box to the flag. Here's a logical approach:

1. Identify the Goal: The goal is to move the box from its current position to the flag. The flag is located in the bottom-right corner of the grid.

2. Box's Position: The box is currently in the center of the grid. To reach the flag, the box needs to be moved horizontally and vertically.

3. Consider the Player's Role: The player can only push the box. The player must stand next to the box to push it. The player's position is to the right of the box, which means the player can push the box to the right.

4. Plan the Moves:

- First, the player needs to push the box to the right. This will move the box one cell to the right.
- Next, the player needs to push the box down. This will move the box one cell down.
- Repeat the process of pushing the box to the right and then down until the box reaches the flag.

5. Count the Moves: Each push to the right and each push down counts as a move. Since the box is in the center and the flag is in the bottom-right corner, the box needs to be pushed 2 cells to the right and 2 cells down. This means a total of 4 pushes are required.
The logical steps involve understanding the grid layout, the movement constraints, and the goal. By planning the sequence of pushes, we can determine the minimum number of moves needed.
</think>

<rethink>
Let's re-evaluate the reasoning based on the visual elements and the logical steps:

1. Visual Confirmation of Box and Flag Positions: The box is clearly in the center of the grid. The flag is in the bottom-right corner. The player is positioned to the right of the box, which is crucial for pushing the box.

2. Movement Analysis: The box needs to be moved 2 cells to the right and 2 cells down to reach the flag. Each push moves the box by 1 cell in the direction of the push. Therefore, to move 2 cells in one direction, 2 pushes are required.

3. Sequence of Moves: First, push the box 2 cells to the right. Then, push the box 2 cells down. This sequence ensures the box moves directly to the flag.

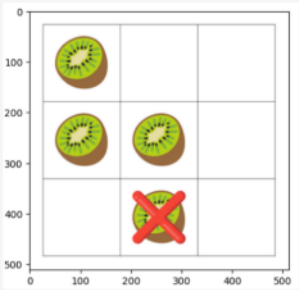
4. Counting the Pushes: The first push moves the box 2 cells to the right. The second push moves the box 2 cells down. The total number of pushes is 2 (right) + 2 (down) = 4.

The visual elements confirm the grid layout and the positions of the box and the flag. The logical steps align with the movement constraints and the goal. The re-evaluation confirms that the minimum number of pushes required is indeed 4.

Thus, the visual evidence supports the reasoning, and the final answer is validated.
</rethink>

The minimum number of pushes required to move the box to the flag is <answer>4</answer>.

Figure 14: The RL-trained model generates detailed and well-structured reasoning for solving the puzzle, thoroughly describing all relevant visual and algorithmic elements without unnecessary repetition. This example illustrates how the $\mathcal{R}_{\text{STRUCTURED}}$ reward function guides the model toward clear, precise, and long multistep reasoning. The key part of the reasoning is bold or highlighted.



Rotting Kiwi Puzzle

<image description>
The image presents a 3x3 grid. Here is the analysis of key elements:
1. Kiwis Representation: Kiwis are depicted with a green outer layer and a brown inner layer. The green inner layer contains black seeds. There are three kiwis visible in the grid.

2. State of Kiwis: The kiwi in the top-left cell is fresh. It is green with black seeds and brown skin. The kiwi in the middle-left cell is also fresh. It shares the same characteristics as the top-left kiwi. The kiwi in the middle-right cell is marked with a red X. This indicates that this kiwi is rotten. It is also green with black seeds and brown skin.

3. Grid Structure and Adjacency: The grid is divided into nine cells. Each kiwi is positioned in a cell. The rotten kiwi in the middle-right cell is adjacent to the fresh kiwi in the middle-left cell.

4. Color and Markings: The rotten kiwi is marked with a red X, indicating its state.

5. Contextual Understanding: The question involves the spread of rot. Any fresh kiwi adjacent to a rotten kiwi will become rotten.

These elements in the grid are crucial for understanding how the kiwis will change over time.

</image description>

<think>
Let's use this image and these elements to reason about the question. Here's a logical breakdown:

1. Understanding the Grid and Initial States: We have a 3×3 grid. The grid contains three fresh kiwis and one rotten kiwi. The rotten kiwi is adjacent to one of the fresh kiwis.

2. Analyzing the Spread Mechanism: Any fresh kiwi adjacent to a rotten kiwi will become rotten. This means the rotten kiwi in the middle-right cell will cause the adjacent fresh kiwi in the middle-left cell to become rotten.

3. Immediate Impact Analysis: Since the rotten kiwi in the middle-right cell is adjacent to the fresh kiwi in the middle-left cell, this fresh kiwi will turn rotten in the next minute.

4. Subsequent Changes: After one minute, the grid will have two rotten kiwis: the original rotten kiwi in the middle-right cell and the newly rotted kiwi in the middle-left cell.

5. Considering Further Minutes: The rotten kiwi in the middle-right cell is also adjacent to the fresh kiwi in the top-left cell. This means that in the next minute, that fresh kiwi will also become rotten.

6. Final State Prediction: If this process continues, every fresh kiwi will eventually become rotten. By analyzing these steps, we can predict that the minimum number of minutes required for all fresh kiwis to become rotten depends on how adjacency drives the spread of rot. Since the rotten kiwi triggers the rotting of adjacent fresh kiwis, the spread will continue until all fresh kiwis are rotten.
</think>

<rethink>
Let's re-examine the grid and the process of rotting kiwis. Here's a detailed breakdown:
1. Initial Observation and Grid Analysis: The grid has three fresh kiwis and one rotten kiwi. The rotten kiwi is located in the middle-right cell and is adjacent to the fresh kiwi in the middle-left cell.

2. Understanding the Spread Mechanism: Any fresh kiwi adjacent to a rotten kiwi will turn rotten. Therefore, the fresh kiwi in the middle-left cell will turn rotten in the next minute.

3. Immediate Change Analysis: After the first minute, the fresh kiwi in the middle-left cell becomes rotten. The grid now has two rotten kiwis and one fresh kiwi.

4. Next Minute Analysis: The rotten kiwi in the middle-right cell is also adjacent to the fresh kiwi in the top-left cell. This fresh kiwi will turn rotten in the next minute.

5. Subsequent Changes: After that minute, the fresh kiwi in the top-left cell becomes rotten. This leaves one remaining fresh kiwi in the bottom row.

6. Final Minute Analysis: The rotten kiwi in the middle-right cell is adjacent to the remaining fresh kiwis in the bottom-left and bottom-center cells. These kiwis will turn rotten in the next minute, and the spread will continue to the bottom-right kiwi. By this reasoning, the process continues until all fresh kiwis are rotten. The grid evolves as follows:
Top-left: Rotten, Middle-left: Rotten, Middle-right: Rotten, Bottom-left: Fresh, Bottom-center: Fresh, Bottom-right: Fresh

7. Final State Prediction: The rotten kiwi in the middle-right cell will cause the fresh kiwis in the bottom-left and bottom-center cells to rot. The rot then propagates to the bottom-right cell. Thus, the minimum number of minutes needed for all fresh kiwis to become rotten is three.
</rethink>

<answer>3</answer>.

Figure 15: The RL-trained model demonstrates structured multistep visual reasoning without any supervised training samples. The reward incentivizes the model to elicit long reasoning and to explain the image in detail, like Kiwi's representation, grid structure, color, and markings. The designed reward encourages long, detailed chains of thought, yielding clear, stepwise, and self-checked explanations.

Prompt for $\mathcal{R}_{\text{FORMAT}}$ reward

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process and then provides the user with the answer.
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>.

Prompt for $\mathcal{R}_{\text{OUTCOME}}$, $\mathcal{R}_{\text{STRUCTURED}}$, $\mathcal{R}_{\text{GRADED}}$, $\mathcal{R}_{\text{PROCESS}}$ rewards

You are a visual reasoning model. Your task is to analyze a visual input and answer the question based on it. Follow this four-stage reasoning framework, using information from each step to inform the next:
<image description>
Begin by providing a description of the key visual elements in the image that can be relevant to the question. Focus on objects, spatial relationships, color, text, or any other patterns that might be important.
</image description>
<think>
Use the image description to reason about the question. What logical steps or background knowledge help interpret the image in context of the question? Lay out general reasoning steps to solve the question without jumping to the final answer.
</think>
<rethink>
Reflect on the image description and the reasoning steps. Rethink and reevaluate whether the visual evidence supports the reasoning. Describe your reevaluation process. Correct or adjust your interpretation to better justify a confident final answer.
</rethink>
<answer>
Provide final answer of the question based on your validated reasoning.
</answer>

Prompt for $\mathcal{R}_{\text{GROUNDED}}$ reward

You are a visual reasoning model. Analyze an image and answer the question using structured reasoning. Begin with a reasoning process enclosed in <think> </think> tags. This should reflect your internal monologue, laying out the logic step by step to answer the question.
Within the <think> tag, include <visual> </visual> tags to describe distinct and meaningful observations from the image. Each <visual> tag must present unique insights—do not repeat content or include vague or empty descriptions. Refine or adjust your reasoning as needed based on what you observe. At the end, provide your final answer in a single <answer> </answer> tag placed outside the <think> block. The response must include at least one <visual> tag within <think>, and exactly one <answer> tag.

Table 10: Prompts for different reward functions. We use the same prompt for the $\mathcal{R}_{\text{OUTCOME}}$, $\mathcal{R}_{\text{STRUCTURED}}$, $\mathcal{R}_{\text{GRADED}}$, and $\mathcal{R}_{\text{PROCESS}}$ rewards, with each function differing only in how the reward is computed. The $\mathcal{R}_{\text{FORMAT}}$ and $\mathcal{R}_{\text{GROUNDED}}$ rewards use separate prompts. All prompts were selected based on trial-and-error and our empirical observations.

Pseudo-code for calculating $\mathcal{R}_{\text{FORMAT}}$ and $\mathcal{R}_{\text{OUTCOME}}$ rewards

```
FUNCTION GradeAnswer(prediction, ground_truth):
    EXTRACT all contents inside <answer> ... </answer>
    IF no answer found:
        RETURN 0
    LET final_answer be the last extracted answer
    IF final_answer matches ground_truth:
        RETURN 1
    ELSE:
        RETURN 0


FUNCTION RewardCalc(prediction, ground_truth, α_t, α_a):
    think_reward = 0

    EXTRACT all contents inside <think> ... </think>
    IF think content found:
        think_reward = α_t

    accuracy_reward = α_a * GradeAnswer(prediction, ground_truth)

    RETURN {
        'overall': think_reward + accuracy_reward,
        'think': think_reward,
        'accuracy': accuracy_reward
    }

# Setting α_t = 0 and α_a = 1 yields in R_outcome reward
```

Pseudo-code for calculating $\mathcal{R}_{\text{STRUCTURED}}$, $\mathcal{R}_{\text{GRADED}}$, and $\mathcal{R}_{\text{PROCESS}}$ rewards

```
FUNCTION SideReward(prediction, tag):
    EXTRACT text inside <tag> ... </tag>
    IF tag not present:
        RETURN 0

    SPLIT text into sentences
    COUNT unique_sentences
    COUNT total_sentences

    smoothed_score <- tanh(unique_sentences)
    duplicate_score <- tanh(total_sentences - unique_sentences)
    RETRUN smooth_score - duplicate_score


FUNCTION RewardCalc(prediction, ground_truth, α_i, α_t, α_r, α_a):
    image_reward <- SideReward(prediction, "image description")
    think_reward <- SideReward(prediction, "think")
    rethink_reward <- SideReward(prediction, "rethink")
    accuracy_reward <- GradeAnswer(prediction, ground_truth)

    overall_reward <- (α_i * image_reward + α_t * think_reward + α_r * rethinkg_reward + α_a
        * accuracy_reward)

    RETURN {
        'overall': overall_reward,
        'image': image_reward,
        'think': think_reward,
        'rethink': rethink_reward,
        'accuracy': accuracy_reward
    }

# the parameters α_i, α_t, α_r, α_a are changed for different reward calculation.
```

Additional details on calculating $\mathcal{R}_{\text{GRADED}}$ reward

```
#Clock-based \rGraded reward
FUNCTION HourScore(pred_hour, true_hour):
    hour_diff = circular distance on 12-hour clock
    IF hour_diff > 2:
        RETURN 0.0
    ELSE:
        RETURN 1 - (hour_diff / 5)

FUNCTION MinuteScore(pred_min, true_min):
    minute_diff = circular distance on 60-minute clock
    IF minute_diff > 10:
        RETURN 0.0
    ELSE:
        RETURN 1 - (minute_diff / 20)

FUNCTION ClockReward(pred_time, true_time):
    CONVERT pred_time to (pred_hour, pred_minute)
    CONVERT true_time to (true_hour, true_minute)

    hour_score = HourScore(pred_hour, true_hour)
    minute_score = MinuteScore(pred_minute, true_minute)

    RETURN 0.5 * hour_score + 0.5 * minute_score

#Numeric \rGraded reward
FUNCTION NumericReward(pred_value, true_value):
    IF conversion to number fails:
        RETURN 0.0
    numeric_diff = abs(pred_value, true_value)

    #if the deviation exceeds this threshold, set the reward to 0. We set it to 5.
    IF numeric_dff > 5.0:
        RETURN 0.0
    ELSE:
        1.0/(1.0+numeric_diff) #if no error, then get full reward; otherwise, get a reward
             score that will linearly decay from 0.5.

FUNCTION GradeAnswer(prediction, ground_truth):
    EXTRACT all contents inside <answer> ... </answer>
    IF no answer found:
        RETURN 0.0

    LET given_answer be the last extracted answer

    IF given_answer and ground_truth are in time format (HH:MM):
        RETURN ClockReward(given_answer, ground_truth)
    ELSE:
        RETURN NumericReward(given_answer, ground_truth)

FUNCTION RewardCalc(prediction, ground_truth, α_i, α_t, α_r, α_a):
    image_reward <- SideReward(prediction, "image description")
    think_reward <- SideReward(prediction, "think")
    rethink_reward <- SideReward(prediction, "rethink")
    continuious_reward <- GradeAnswer(prediction, ground_truth)

    overall_reward <- (α_i * image_reward + α_t * think_reward + α_r * rethinkg_reward + α_a
        * \rGraded_reward)

    RETURN {
        'overall': overall_reward,
        'image': image_reward,
        'think': think_reward,
        'rethink': rethink_reward,
        'accuracy': continuious_reward
    }
```

Pseudo-code for calculating $\mathcal{R}_{\text{GROUNDED}}$ reward

```
FUNCTION CountUniqueVisualTags(prediction):
    EXTRACT content inside <think>
    FIND all <visual> ... </visual> tags
    REMOVE empty or duplicate visual descriptions
    COUNT unique_visual_tags
    COUNT duplicate_visual_tags
    RETURN tanh(unique_visual_tags) - tanh(duplicate_visual_tags)

FUNCTION \rGroundedReward(prediction, ground_truth, α_t, α_v, α_a):
    think_reward = 0
    EXTRACT all contents inside <think> ... </think>
    IF think content found:
        think_reward = α_t

    vision_reward <- CountUniqueVisualTags(prediction)
    accuracy_reward <- GradeAnswer(prediction, ground_truth)

    overall_reward <- (α_v * vision_reward + α_t * think_reward + α_a * accuracy_reward)

    RETURN {
        'overall': overall_reward,
        'vision': vision_reward,
        'think': think_reward,
        'accuracy': accuracy_reward
    }
```

Table 11: Pseudo-code for reward design used in the reinforcement learning framework. The above codes summarize the reward formulations used in our experiments. The top block shows the $\mathcal{R}_{\text{FORMAT}}$ and $\mathcal{R}_{\text{OUTCOME}}$ rewards, where answer correctness is computed by GradeAnswer and formatted reasoning is optionally encouraged via a think-token reward weighted by α_t (setting α_t=0 yields $\mathcal{R}_{\text{OUTCOME}}$). The middle block presents $\mathcal{R}_{\text{STRUCTURED}}$, $\mathcal{R}_{\text{GRADED}}$, and $\mathcal{R}_{\text{PROCESS}}$ rewards that combine signals from image descriptions, reasoning (`<think>`), revisions (`<rethink>`), and answer accuracy using weights $\alpha_i, \alpha_t, \alpha_r, \alpha_a$. The bottom block details the $\mathcal{R}_{\text{GRADED}}$ and $\mathcal{R}_{\text{GROUNDED}}$ rewards. $\mathcal{R}_{\text{GRADED}}$ reward assigns partial credit based on clock-time or numeric deviation, and visual reward incentivizes grounded visual references via unique `<visual>` tags.