THINK IN GRAPHS: DISENTANGLED GRAPH-CENTRIC MULTIMODAL REASONING VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has significantly improved reasoning capabilities in large language models, yet extending these advances to vision-language models remains challenging due to insufficient modeling of spatial and logical relations among visual entities. Existing methods rely on isolated visual entities and final-answer supervision, which causes reasoning to drift toward language priors rather than visual relational structures. We propose GRATS, a two-stage multimodal reasoning framework that consists of constructing relational graphs from images and performing graph-centric chain-of-thought reasoning over the generated graphs. We further introduce a novel RL method with stage-specific credit assignment to optimize the model's graph construction and graph reasoning capabilities simultaneously. Experiments on seven multimodal reasoning benchmarks show that GRATS consistently improves performance on both Physical Perception and Diagram & Quantitative Reasoning tasks. Code is available at: https://anonymous.4open.science/r/GRATS.

1 INTRODUCTION

Reinforcement learning (RL) has proven effective Ouyang et al. (2022); Lightman et al. (2024); Guo et al. (2025) in improving reasoning capabilities in large language models (LLMs). Compared with LLMs, vision-language models (VLMs) additionally require integrating and analyzing visual information during reasoning Acharya et al. (2019); Li et al. (2023); Liu et al. (2023b); Peng et al. (2024). However, existing multimodal reasoning methods lack explicit modeling of spatial and logical relations among visual entities Krishna et al. (2017); Lu et al. (2016); Xu et al. (2017); Zellers et al. (2018); Yang et al. (2018). Consequently, models have to rely on language priors Goyal et al. (2017) rather than visual facts during reasoning, which frequently causes hallucination. Therefore, tailored RL frameworks for enhancing spatial and logical multimodal reasoning are urgently needed.

Enhancing with RL faces three key challenges. First, existing visual perception methods usually produce discrete entities, ignoring spatial and logical information among entities, which is essential in multimodal reasoning Peng et al. (2024); Chen et al. (2023); Yang et al. (2023). Second, even with structured visual information, models often resist using such information and degenerate to language-only reasoning. This implies that models' ability to "think with images" needs to be specially strengthened Wei et al. (2022); Zhang et al. (2023); Yi et al. (2018); Hudson & Manning (2019). Third, multimodal reasoning usually involves multi-step reasoning processes, while the outcome rewards only provide guidance on the final answers. This gap makes vanilla RL methods less effective in optimizing VLMs Lightman et al. (2024); Liu et al. (2025b); Huang et al. (2025).

Prior works have demonstrated the value of RL in multimodal reasoning, yet their improvements remain limited. Early methods enhance visual perception of VLMs by forcing them to involve visual tokens in the reasoning process Alayrac et al. (2022); Li et al. (2023); Liu et al. (2023b); Peng et al. (2024); Chen et al. (2023); Yang et al. (2023). Recent works further improve multimodal reasoning by encouraging models to repeatedly reference visual tokens or replay visual information during reasoning Wang et al. (2025b); Yang et al. (2026). Although these methods improve the use of visual tokens, models have not actually learned to perceive spatial and logical information within images, and how to fully exploit this information during reasoning Zhang et al. (2023); Wang

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al. (2025b); Yang et al. (2026); Lee et al. (2024); Liu et al. (2025a). Moreover, existing RL methods typically optimize only final answers, making it difficult to distinguish perception errors from reasoning errors Hudson & Manning (2019); Lightman et al. (2024).

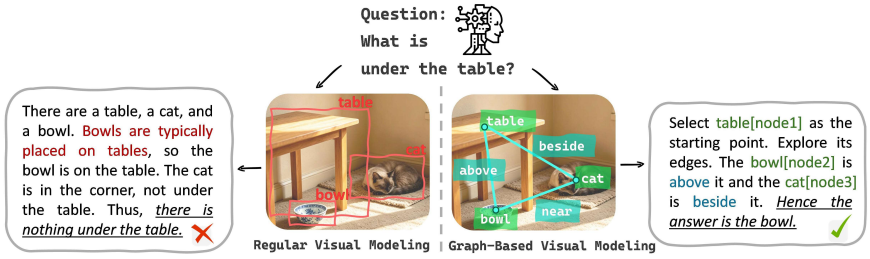


Figure 1: Comparison between regular visual modeling and graph-based visual modeling on spatial reasoning. **Left**: Regular visual modeling detects isolated objects with bounding boxes but lacks explicit relational structure, causing the model to rely on language priors (e.g., bowls are typically placed on tables) and produce an incorrect answer. **Right**: Graph-based visual modeling represents objects as nodes connected by spatial predicates, enabling stepwise graph-based reasoning over structured visual relations and more effective use of visual evidence.

In this paper, we present **GRATS**, a novel multimodal learning framework that enhances graph-level perception and reasoning capabilities of VLMs. GRATS explicitly models visual information as a graph, and guides the VLMs to perform reasoning under graph-centric chain-of-thoughts. In this way, the reasoning process is more reliable and explainable. Figure 1 illustrates the importance of graph modeling in multimodal reasoning. Specifically, GRATS consists of a perception stage and a reasoning stage. First, in the **perception stage**, the VLM detects visual entities and models their spatial and logical relations, organizing them into a relational graph. Second, in the **reasoning stage**, the same VLM performs a multi-step graph-centric reasoning process based on the constructed graph. The reasoning process continuously refers to the graph following a graph-centric chain-of-thought, thus avoiding drift from visual information to language priors. To enhance both the graph perception and reasoning simultaneously, we develop a novel RL method with a tailored credit assignment mechanism to distinguish whether failures are caused by graph construction or reasoning. Our main contributions are summarized as follows.

- We propose a two-stage graph-centric multimodal reasoning framework that explicitly models visual entities and their spatial and logical relations as a graph, guiding VLMs to perform explainable, graph-centric chain-of-thought reasoning to reduce hallucination.
- We introduce a tailored RL method with stage-specific credit assignment to distinguish errors in graph construction from reasoning flaws, improving perception and reasoning.
- We evaluate GRATS on seven multimodal reasoning benchmarks covering Physical Perception and Diagram & Quantitative Reasoning, and experimental results demonstrate consistent improvements over strong baselines.

2 Related Works

Visual Representation in VLMs. Accurate visual representation is essential for VLMs. Early systems, including BLIP-2 Li et al. (2023), Flamingo Alayrac et al. (2022), and LLaVA Liu et al. (2023b), typically rely on pretrained vision encoders such as CLIP Radford et al. (2021) to extract grid-based features. These features are efficient and effective for scene-level perception, but their coarse granularity limits fine-grained spatial and relational modeling. Later work improves grounding with object-centric models such as Grounding DINO Liu et al. (2024) and SAM Kirillov et al. (2023), or by injecting coordinate tokens into VLMs, as in Shikra Chen et al. (2023) and G-LLaVA Gao et al. (2025), among others. However, despite improving object-level recognition, these methods largely overlook inter-entity relationships. Scene graph generation methods, such as IMP Xu et al. (2017), Neural Motifs Zellers et al. (2018), and unbiased SGG Tang et al. (2020), explicitly represent pairwise subject-predicate-object triplets. Nevertheless, in many challenging scenarios, such binary relations are often insufficient for capturing the increasingly complex relational structures. To bridge this gap, we introduce a structured graph representation that organizes multi-object dependencies and supports multi-step reasoning over complex scenes.

Mitigating Hallucination in Multimodal Reasoning. CoT Wei et al. (2022) prompting has substantially improved text reasoning, motivating its extension to VLMs. Multimodal-CoT Zhang et al. (2023), for example, combines visual and textual inputs for ScienceQA Lu et al. (2022). Yet many early paradigms depend on offline, static visual features, making long-horizon reasoning prone to language priors rather than visual evidence. Recent work addresses this issue from several directions. Tool-augmented systems such as Chameleon Lu et al. (2023b), MAPS, and related tool-based methods Zhang et al. (2026) impose external constraints, but tool outputs are often converted into text, leaving the final reasoning process vulnerable to linguistic commonsense. VGR Wang et al. (2025b) and ViGoR Yan et al. (2024) introduce visual playback or feedback at intermediate steps Yang et al. (2022); Wang et al. (2025a). However, such feedback is reactive and lacks an explicit scene structure, so reasoning can still drift from visual evidence. PSGP Li (2026) models robust argument paths probabilistically, but underuses structured visual information. In contrast, our method grounds each reasoning step in a structured graph, leading to reliable, explainable reasoning.

RL for VLM Reasoning. RL methods, such as GRPO, Shao et al. (2024b) have demonstrated remarkable success in text-based tasks, prompting a surge in adapting RL for VLMs Yu et al. (2024); Sun et al. (2024). Unlike pure text, multimodal reasoning relies on both textual logic and visual perception, and outcome-only rewards provide coarse guidance for complex visual tasks. To refine these signals, recent studies focus on specific components of the reasoning chain. For reasoning, Process Reward Models (PRMs) Lightman et al. (2024) provide fine-grained logical supervision. Regarding perception, approaches like ViGoR Yan et al. (2024) and R1-SGG Chen et al. (2025) integrate structural evidence extraction into their optimization objectives. Nevertheless, existing methods still couple perception and reasoning rewards, making it difficult to identify whether failures stem from perception errors or reasoning errors. To address this, our framework separates perception and reasoning signals for more accurate and reliable visually grounded supervision.

3 PRELIMINARIES

Group-Based Policy Optimization. Group-based policy optimization methods optimize policies using relative comparisons among multiple sampled responses, avoiding value-function estimation. Representative methods include Group Relative Policy Optimization (GRPO) and its extensions such as Dynamic Sampling Policy Optimization (DAPO), which improves training efficiency and exploration via adaptive sampling and group-wise optimization strategies.

Given a prompt x, the policy π_θ samples a group of G responses $\{y_i\}_{i=1}^{G}$ with rewards $\{r_i\}_{i=1}^{G}$. The advantage is estimated by group-relative normalization, $\hat{A}_i = \frac{r_i - \mu_r}{\sigma_r}$, where μ_r and σ_r are the mean and standard deviation of rewards within the group. The policy is optimized using a PPO-style clipped objective with KL regularization:

$$\ell_{i,t}(\theta) = \min\Big(\rho_{i,t}\hat{A}_i,\ \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_i\Big). \tag{1}$$

$$\mathcal{L}_{\text{Group}} = \mathbb{E}_{x,\{y_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\ell_{i,t}(\theta) - \beta D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\right]. \tag{2}$$

$$\text{where}\quad \rho_{i,t} = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})}.$$

Problem Statement. We consider an agentic reinforcement learning approach for multimodal reasoning, where the agent receives an image I and a question q, and is required to generate a final answer a. The model is trained to learn a policy $\pi_\theta(a \mid I, q)$ that directly maps multimodal inputs to the correct output through sequential reasoning. The optimization objective is to maximize the expected reward:

$$\max_\theta\ \mathbb{E}_{(I,q)}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|I,q)}\big[R(a)\big]\right]. \tag{3}$$

Despite recent advances, applying RL to multimodal reasoning in VLMs remains challenging. The challenges come from visual perception, reasoning and learning aspects. This motivates us to de-

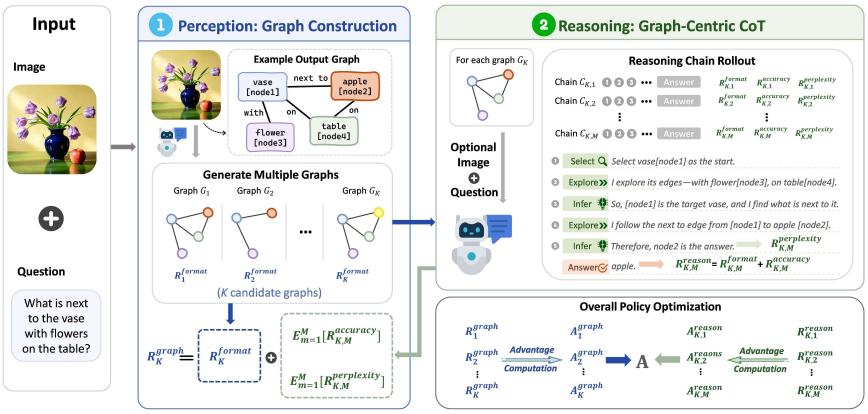


Figure 2: Overview of GRATS. Given an image–question input, GRATS samples K relational graphs in the **perception stage** and M stepwise reasoning trajectories per graph in the **reasoning stage**. Graph-level rewards encourage valid relational structure and downstream reasoning utility; reasoning-level rewards emphasize structure-consistent graph operations and answer correctness. Normalized advantages from both stages are applied to phase-specific tokens for policy credit assignment.

velop a graph-centric multimodal reasoning and learning approach, which mimics how humans understand and think with images.

4 METHODS

Motivation. Existing multimodal reasoning methods usually represent visual information as isolated entities, limiting structured reasoning over visual relations. By contrast, when humans see an image, they usually construct an abstract graph in their mind, shortly memorizing key entities and their relationships. Motivated by this, we aim to empower VLMs with the ability to explicitly perceive and think with abstract graphs, rather than raw visual tokens.

Overview. As illustrated in Figure 2, we propose GRATS, a two-stage RL framework for multimodal reasoning. GRATS decomposes multimodal reasoning into relational structure construction and graph-centric reasoning. With separate RL advantages for the two stages, GRATS enables explicit visual relation modeling and improved credit assignment for reliable multimodal reasoning.

4.1 RELATIONAL GRAPH CONSTRUCTION

Given an image-question input $x = (I, Q)$, we prompt a base MLLM to construct a visual relational graph in two sequential steps. First, the MLLM generates a set of relational triples in the form $(subject, relation, object)$, describing pairwise logical or spatial relations between visual entities in the image. These triples capture local interactions between objects. Then, the model organizes the generated triples into a graph representation G for downstream reasoning: $G = \text{MLLM}(I, Q)$. In the graph G, nodes correspond to visual entities with unique identifiers (e.g., $vase[node1]$), while edges represent explicit logical or spatial relations induced from the generated triples. In Figure 2, the image contains a vase with flowers next to an apple, where both objects are on a table. The model identifies entities including the vase, apple, flower, and table, and constructs their relations accordingly. Unlike ordinary graphs that connect nodes using generic similarity or co-occurrence signals, the proposed relational graph explicitly models logical and spatial relations between objects, making the graph more visually grounded and better for multi-step reasoning.

4.2 GRAPH-CENTRIC COT

To reduce reliance on language priors in multimodal reasoning, we ground the reasoning process on an explicitly constructed relational graph instead of unconstrained latent inference. Given a relational graph G_i, question Q and optional image, the model samples: $y_i \sim \pi_\theta(\cdot \mid G_i, Q)$, where $y_i =$

$\{reasoning, answer\}$. The reasoning field consists of graph operations: $O_i = \{o_1, o_2, \ldots, o_T\}$, where each step explicitly interacts with graph nodes and relations through three basic operations: *Select*, *Explore*, and *Infer*.

Select retrieves graph nodes that satisfy semantic or structural conditions. This operation serves to initialize the reasoning process from visually grounded entities relevant to the question, preventing the model from directly relying on language priors during early reasoning stages. For example, given the question "What object is under the table?", the model first selects the node corresponding to the table instead of implicitly inferring candidate answers from statistical language correlations.

Explore denotes moving from a given node to a connected node through a single relation edge defined by a predicate. Unlike standard CoT reasoning that implicitly models relations in latent space, Explore explicitly follows graph predicates to discover relational neighbors. For example, given the question " What animal is in the car? ", after selecting `car[node1]`, the model follows the predicate `in` to explore the connected node `dog[node2]` rather than relying on language priors about common object arrangements. This operation enforces reasoning over explicit visual relations and reduces potential reasoning drift from unreliable language priors.

Infer performs intermediate verification or final decisions based on the explored subgraph. The operation aggregates relational evidence from previous steps to verify attributes, validate conditions, or produce the final answer. By separating exploration from inference, the model is encouraged to explicitly gather relational evidence before generating conclusions, improving overall reasoning consistency and interpretability.

4.3 Disentangled RL for Policy Optimization

In the policy optimization stage, GRATS optimizes relational graph construction and graph-centric reasoning with disentangled objectives. We design dedicated rewards for generated graphs and reasoning trajectories, and perform stage-specific grouping and advantage estimation for fine-grained token-level credit assignment. Reward designs and advantage computation are detailed below.

Graph-level reward design. We assign a graph-level reward to each candidate graph:

$$R_i^{\text{graph}} = \lambda_{\text{fmt}} F_i^{\text{graph}} + \lambda_{\text{ppl}} P_i^{\text{reason}} + \lambda_{\text{ans}} R_{\text{leak}}(c_i) S_i^{\text{reason}}. \tag{4}$$

where

$$F_i^{\text{graph}} = \text{Norm}\Big(R_{\text{json}}(c_i) + R_{\text{struct}}(c_i) + R_{\text{node}}(c_i)\Big) \tag{5}$$

denotes the normalized graph-format score. Here, R_{json} checks JSON validity, R_{struct} evaluates graph completeness, and R_{node} enforces node consistency. To encourage graphs that better support downstream reasoning, we introduce two auxiliary signals from second-stage reasoning trajectories. S_i^{reason} measures average answer correctness conditioned on graph G_i. A leakage gate $R_{\text{leak}} \in \{0, 1\}$ suppresses graphs that explicitly expose the ground-truth answer. P_i^{reason} denotes the average perplexity-based reward, reflecting reasoning confidence and consistency. The perplexity signal is only used for graph-level optimization.

Reasoning-level reward design. Given the constructed relational graph and reasoning trajectory, we compute the reasoning reward:

$$R_i^{\text{reason}} = \lambda_{\text{gpt}} S_i + \lambda_{\text{fmt}} F_i^{\text{reason}}, \tag{6}$$

where

$$F_i^{\text{reason}} = \text{Norm}\Big(R_{\text{json}}(c_i) + R_{\text{struct}}(c_i) + R_{\text{op}}(c_i) + R_{\text{node}}(c_i)\Big). \tag{7}$$

denotes the normalized reasoning-format score, and S_i is a GPT-4o-based correctness score. R_{op} evaluates whether the reasoning trajectory follows valid graph operations. R_{json} checks JSON validity, R_{struct} evaluates CoT completeness, and R_{node} enforces node consistency.

Two-stage grouping. Given an image-question input $x = (I, Q)$, we first sample K relational graphs from the current policy: $\{G_1, \ldots, G_K\} \sim \pi_\theta(\cdot|x)$. For each graph G_i, we further sample M reasoning trajectories conditioned on the generated graph: $\{r_{i,1}, \ldots, r_{i,M}\} \sim \pi_\theta(\cdot|x, G_i)$. Further

sensitivity analyses of the perplexity reward weight and the number of generated graphs are provided in Appendix D.

Graph-level advantage. We compute graph-level rewards R_i^{graph} and normalize them within the graph candidate group:

$$A_i^{\text{graph}} = \frac{R_i^{\text{graph}} - \frac{1}{K}\sum_{j=1}^{K} R_j^{\text{graph}}}{\text{std}(\{R_j^{\text{graph}}\}_{j=1}^{K}) + \delta}. \tag{8}$$

This advantage measures the relative quality of graph construction under the same input.

Reasoning-level advantage. For each reasoning trajectory $r_{i,m}$ conditioned on graph G_i, we compute reasoning rewards $R_{i,m}^{\text{reason}}$ and normalize them within the reasoning group:

$$A_{i,m}^{\text{reason}} = \frac{R_{i,m}^{\text{reason}} - \frac{1}{M}\sum_{n=1}^{M} R_{i,n}^{\text{reason}}}{\text{std}(\{R_{i,n}^{\text{reason}}\}_{n=1}^{M}) + \delta}. \tag{9}$$

This signal measures the relative reasoning quality under a fixed graph structure.

Overall policy optimization. We integrate graph-level and reasoning-level signals into a unified optimization objective. Graph-level advantages are applied to tokens generated during graph construction, while reasoning-level advantages are applied to tokens generated during graph-centric reasoning. This design enables phase-specific optimization within a single policy update. We denote $\pi_{\theta_{\text{old}}}$ as the sampling policy and π_θ as the updated policy. Each candidate sample is represented as a token sequence $c_i = (c_{i,1}, \ldots, c_{i,T_i})$, including both graph construction outputs and reasoning trajectories. For each token position t, the importance ratio is:

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(c_{i,t} \mid x, c_{i,1:t-1})}{\pi_{\theta_{\text{old}}}(c_{i,t} \mid x, c_{i,1:t-1})}. \tag{10}$$

We adopt the clipped surrogate objective following GRPO. For each token, we combine graph-level and reasoning-level advantages using phase-specific masks:

$$A_{i,t} = M_{i,t}^{\text{graph}} A_i^{\text{graph}} + M_{i,t}^{\text{reason}} A_i^{\text{reason}}, \tag{11}$$

where $M_{i,t}^{\text{graph}}$ and $M_{i,t}^{\text{reason}}$ indicate whether token t belongs to graph construction or reasoning generation. The overall objective is:

$$\begin{aligned} L(\theta) = -\mathbb{E}_x \Bigg[\frac{1}{N}\sum_{i=1}^{N} \frac{1}{T_i}\sum_{t=1}^{T_i} \min\Big(\rho_{i,t}(\theta) A_{i,t}, \\ \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big) A_{i,t}\Big) \Bigg] + \beta D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big), \end{aligned} \tag{12}$$

where ϵ is the clipping coefficient, π_{ref} denotes the reference policy, and β controls the KL regularization strength. By decoupling graph construction and reasoning into separately normalized learning signals, GRATS enables explicit credit assignment to both perception and reasoning stages.

5 Experiments

In this section, we evaluate GRATS on widely used multimodal reasoning benchmarks to answer the following research questions (RQs).

- **RQ1:** Can GRATS outperform existing multimodal reasoning methods across different benchmark categories?
- **RQ2:** How do the proposed graph construction and graph-centric reasoning modules affect the learning dynamics?
- **RQ3:** Does GRATS encourage stronger utilization of visual evidence and persistent graph-centric reasoning during optimization?
- **RQ4:** What is the computational overhead of GRATS compared with standard GRPO-based optimization?

Table 1: Performance comparison of GRATS with representative closed-source and open-source MLLMs under different settings on answer accuracy (ACC, %). Best scores are bold; second-best scores are underlined. Δ over Qwen3-VL denotes the gain over the Qwen3-VL model at the same parameter scale; Q3-VL abbreviates Qwen3-VL. $^{+}$ marks an official-report score.

	Physical Perception			Diagram and Quantitative Reasoning			Comprehensive Test
Model	**VSR**	**TallyQA**	**GQA**	**ChartQA**	**M.Vista**	**M.Vision**	**MME**
Closed-Source Models							
GPT-4o	62.3	38.9	52.2	76.0	60.6	30.4^{+}	83.5
Gemini-2.5-Pro	64.0	49.8	60.5	84.1	54.1	73.3^{+}	92.9
Open-Source Models (8B)							
Q3-VL	<u>63.0</u>	55.9	<u>77.0</u>	36.5	20.0	14.7	81.8
Q3-VL+DAPO	20.4	<u>74.1</u>	71.3	<u>73.5</u>	43.2	<u>22.9</u>	85.2
Q3-VL+GRPO	56.7	71.1	75.9	71.6	<u>51.9</u>	10.5	<u>86.5</u>
Q3-VL+**GRATS**	**78.9**	**75.0**	**78.8**	**77.9**	**58.8**	**23.8**	**90.3**
Δ *over Q3-VL*	+15.9%	+19.1%	+1.8%	+41.4%	+38.8%	+9.1%	+8.5%
Open-Source Models (4B)							
Q3-VL	18.4	37.6	50.2	31.3	16.7	5.6	<u>78.0</u>
CoT	12.8	8.0	2.6	4.5	3.6	7.4	46.6
1-shot ICL	<u>49.0</u>	<u>68.0</u>	48.0	68.5	49.2	15.1	<u>78.0</u>
Few-shot FT	20.0	64.0	<u>71.0</u>	<u>74.3</u>	45.6	8.0	50.0
Q3-VL+DAPO	47.2	64.0	70.0	72.6	<u>50.4</u>	17.9	**80.0**
Q3-VL+GRPO	47.7	60.0	63.3	70.7	48.3	<u>19.1</u>	74.3
Q3-VL+**GRATS**	**59.6**	**68.8**	**75.9**	**76.7**	**58.0**	**19.3**	**80.0**
Δ *over Q3-VL*	+41.2%	+31.2%	+25.7%	+45.4%	+41.3%	+13.7%	+2.0%

Datasets. Experiments are conducted on seven public multimodal reasoning benchmarks, including VSR Liu et al. (2023a), TallyQA Acharya et al. (2019), GQA Hudson & Manning (2019), ChartQA Masry et al. (2022), MathVista Lu et al. (2023a), MathVision Wang et al. (2024), and MME Fu et al. (2025). These datasets evaluate different reasoning abilities, ranging from physical perception and spatial understanding to chart interpretation and mathematical reasoning. Additional preprocessing details are provided in Appendix A.

Baselines. We compare GRATS with representative VLM training paradigms under both 8B and 4B settings across different evaluation scenarios. For the 8B setup, we use Qwen3-VL-8B as the backbone and mainly compare with RL-based baselines, including Qwen3-VL+DAPO and Qwen3-VL+GRPO. For the 4B setup, we further include commonly used non-RL methods, such as CoT prompting, one-shot ICL, and few-shot fine-tuning, together with RL baselines including Qwen3-VL+DAPO and Qwen3-VL+GRPO. We additionally report strong proprietary models, including GPT-4o and Gemini-2.5-Pro, for reference.

Evaluation Metrics. We adopt answer accuracy as the primary evaluation metric. Following recent multimodal reasoning works, evaluation is performed using a GPT-based judging protocol instead of exact-match comparison, since generated answers may differ in wording while remaining semantically equivalent and correct. Specifically, GPT-4o is used to assess whether the predicted answer matches the reference answer. Final results are reported as average accuracy per benchmark.

5.1 Performance Comparison with Baselines (RQ1)

Table 1 reports the overall comparison results. While GRPO improves multimodal reasoning through answer-level reinforcement learning, it provides limited guidance for intermediate reasoning processes. In contrast, GRATS explicitly decomposes reasoning into graph construction and graph-centric reasoning stages, enabling stage-specific optimization for structured multimodal reasoning. Under the 8B setting, GRATS consistently outperforms Qwen3-VL+GRPO on all seven benchmarks, including VSR (+22.2), TallyQA (+3.9), GQA (+2.9), ChartQA (+6.3), MathVista (+6.9), MathVision (+13.3), and MME (+3.8). Notably, GRATS achieves strong performance on both concrete Physical Perception tasks and more abstract Diagram and Quantitative Reasoning benchmarks, demonstrating effective generalization across different multimodal reasoning scenar-

ios. Similar trends are observed in the 4B setting, where GRATS-4B consistently outperforms Qwen3-VL+GRPO-4B across all benchmarks. Overall, these results demonstrate the effectiveness of graph-centric reasoning and two-stage reinforcement learning for multimodal reasoning.

5.2 Ablation Studies (RQ2)

To analyze the impact of two-stage training on optimization dynamics, we conduct ablation studies on the VSR dataset and track a normalized training reward that jointly measures answer correctness and reasoning-format validity across training iterations. As shown in Figure 3, a consistent trend can be observed: removing Stage-1 (w/o Stage-1, only reasoning) yields the lowest performance and slowest convergence, indicating that the absence of explicit relational graph construction degrades the quality of structured visual representation and thus hinders subsequent reasoning effectiveness; removing Stage-2 (w/o Stage-2, graph supervision only) achieves intermediate performance but fails to effectively translate structural learning into high-quality reasoning, suggesting that structure learning alone is insufficient for complex problem solving; In comparison, the full model overall achieves the best convergence speed and the highest final performance throughout training.

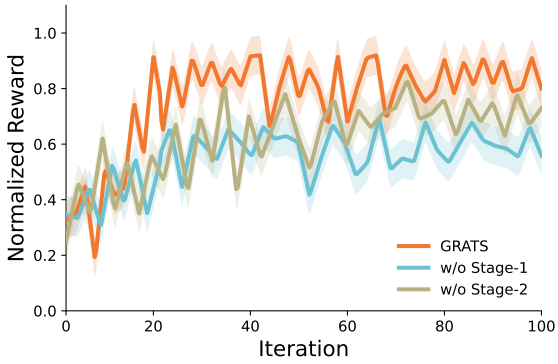


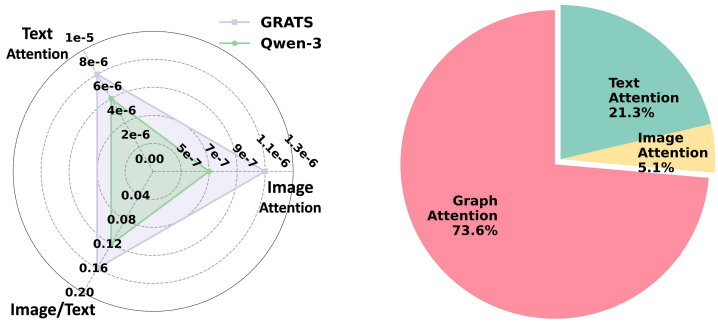
Figure 3: Ablation training curves on VSR. The y-axis reports normalized reward combining answer correctness and reasoning-format validity. w/o Stage-1 (reasoning only) performs worst; w/o Stage-2 (graph supervision only) is intermediate; full GRATS converges fastest and achieves the highest final reward.

To further verify the contribution of each stage, we additionally evaluate two ablated variants on downstream benchmarks. Detailed ablation results are provided in Appendix E. The ablation results show that removing either graph construction or graph-centric reasoning consistently degrades performance. Specifically, removing graph construction causes larger performance drops, with decreases of **8.7**, **5.6**, and **8.3** points on TallyQA, ChartQA, and MathVista, respectively, highlighting the importance of explicit relational visual representations. Removing graph reasoning also leads to consistent degradation across all benchmarks, reducing performance by **5.5**, **4.3**, and **3.8** points on the same benchmarks, demonstrating that constructing structured graphs alone is insufficient for complex multimodal reasoning. These results complement the training dynamics in Figure 3, demonstrating that both structured visual representation and graph-grounded reasoning are necessary for effective multimodal reasoning.

Overall, Stage-1 provides structured visual representations that serve as reliable semantic carriers for reasoning, while Stage-2 performs stepwise reasoning over the constructed structure. Their synergy leads to more stable training dynamics and superior final performance.

5.3 Per-Token Attention Analysis (RQ3)

As shown in Figure 4a, GRATS allocates substantially more attention to visual tokens during graph construction compared with Qwen3. Specifically, image attention increases from 7.35×10^{-7} to 1.17×10^{-6} (+59.2%). Moreover, the image-to-text attention ratio rises from 0.121 to 0.153 (+26.4%), indicating that GRATS relies more heavily on visual evidence rather than textual priors during structured perception. These results suggest that graph-oriented optimization improves visual grounding in the perception stage. Furthermore, as shown in Figure 4b, during the late stage of training, GRATS allocates 2.69×10^{-5} attention to the generated graph, compared with 7.79×10^{-6} for text input and only 1.86×10^{-6} for image features, corresponding to 73.6%, 21.3%, and 5.1% of total attention, respectively. This result shows that GRATS continuously references the graph generated in the first stage throughout reasoning, rather than gradually ignoring structured repre-



(a) (b)

Figure 4: Per-token attention analysis. (a) GRATS vs. Qwen3 during graph construction: higher image attention and image-to-text ratio, indicating improved visual grounding. (b) Late-stage reasoning attention: graph, text, and image tokens account for 73.6%, 21.3%, and 5.1% of attention, respectively, demonstrating persistent graph-centric reasoning over the constructed relational graph.

sentations during optimization. Even in the later training stage, the relational graph remains the dominant reasoning carrier, demonstrating persistent graph-centric reasoning behavior.

5.4 Computational Cost Analysis (RQ4)

To demonstrate the efficiency of our method, we decompose the training process into several distinct components. The time breakdown in Figure 5 shows that the cost is overwhelmingly dominated by generation. Phase-1 and Phase-2 rollouts take 596.69s and 543.81s, respectively, which together account for nearly all of the iteration time. The remaining shared components are much cheaper: the policy update takes 24.99s, while old-policy and reference-policy probability computation take 5.40s and 4.86s. By comparison, the extra computation required by our method is almost negligible. Computing the Phase-1 and Phase-2 advantages takes only 0.00160s and 0.00491s, respectively. This is several orders of magnitude smaller than the rollout cost and accounts for less than 0.001% of the measured iteration time. These results indicate that our method improves the training signal through phase-wise advantage estimation without introducing meaningful computational overhead in the overall process.

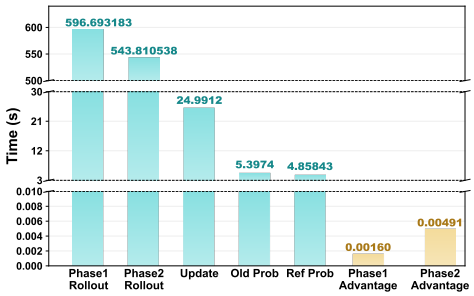


Figure 5: We report the average time of each component in one training iteration. Blue bars correspond to the standard GRPO-style optimization pipeline, while orange bars denote the additional phase-wise advantage computation introduced by our method. Two broken y-axes are used for readability.

6 Conclusions

The goal of multimodal reasoning is to better leverage visual evidence for more reliable answer generation, but existing RL methods remain largely answer-level, limiting structured visual reasoning and effective credit assignment across reasoning stages. We propose **GRATS**, a two-stage RL framework that explicitly builds relational graphs and performs stepwise reasoning over them with decoupled learning signals. Experiments on multiple benchmarks show consistent overall improvements, demonstrating the effectiveness of graph-centric reasoning and two-stage reinforcement learning framework presented here.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and literature search. The authors take responsibility for the final manuscript, including its text, references, and claims.

REPRODUCIBILITY STATEMENT

The method and reward definitions are provided in the main text. Appendix A reports the experiment setup and training data construction; Appendices B–F provide benchmark details, model prompts, sensitivity analyses, stage-wise ablation results, and qualitative examples.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8076–8084, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

Zuyao Chen, Jinlin Wu, Zhen Lei, Marc Pollefeys, and Chang Wen Chen. Compile scene graphs with reinforcement learning. *arXiv preprint arXiv:2504.13617*, 2025.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Xinze Guan, and Xin Wang. Grit: Teaching mllms to think with images. *Advances in Neural Information Processing Systems*, 38:116522–116543, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d79a27cf2772fe00be7f341efc0eb517-Paper-Datasets_and_Benchmarks_Track.pdf.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. In *International Conference on Learning Representations*, volume 2025, pp. 3490–3511, 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Junlin Lee, Yequan Wang, Jing Li, and Min Zhang. Multimodal reasoning with multimodal knowledge graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10767–10782, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Zichao Li. Probabilistic scene graph prompting: Uncertainty-aware structured reasoning in multimodal llms. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 1695–1702, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023a. doi: 10.1162/tacl_a_00566. URL https://aclanthology.org/2023.tacl-1.37/.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023b.

Junming Liu, Siyuan Meng, Yanting Gao, Song Mao, Pinlong Cai, Guohang Yan, Yirong Chen, Zilin Bian, Ding Wang, and Botian Shi. Aligning vision to language: Annotation-free multimodal knowledge graph construction for enhanced llms reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 981–992, 2025a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2034–2044, 2025b.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pp. 852–869. Springer, 2016.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023a.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023b.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding multimodal large language models to the world. In *International Conference on Learning Representations*, volume 2024, pp. 51575–51598, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13088–13110, 2024.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3716–3725, 2020.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*, 2025a.

Jiacong Wang, Zijian Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, et al. Vgr: Visual grounded reasoning. *arXiv preprint arXiv:2506.11991*, 2025b.

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. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

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. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5410–5419, 2017.

Siming Yan, Min Bai, Weifeng Chen, Xiong Zhou, Qixing Huang, and Li Erran Li. Vigor: Improving visual grounding of large vision language models with fine-grained reward modeling. In *European Conference on Computer Vision*, pp. 37–53. Springer, 2024.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

Li Yang, Yan Xu, Chunfeng Yuan, Wei Liu, Bing Li, and Weiming Hu. Improving visual grounding with visual-linguistic verification and iterative reasoning. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9499–9508, 2022.

Shuo Yang, Yuwei Niu, Yuyang Liu, Yang Ye, Bin Lin, and Li Yuan. Look-back: Implicit visual re-focusing in mllm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11694–11702, 2026.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13807–13816, 2024.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5831–5840, 2018.

Shutong Zhang, Dylan Zhou, Yinxiao Liu, Yang Yang, Huiwen Luo, and Wenfei Zou. Tool-mcot: Tool augmented multimodal chain-of-thought for content safety moderation. *arXiv preprint arXiv:2604.06205*, 2026.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

A Experiment Details

A.1 Experiment Setup

We use the AdamW optimizer with a learning-rate scheduler and set the learning rate to 3×10^{-6}. Experiments are conducted on 4 NVIDIA A800 GPUs using DeepSpeed ZeRO-3 optimization. Training is based on the Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct models, with BF16 precision and gradient checkpointing enabled. We use a per-device batch size of 6 with a gradient accumulation step of 2, resulting in an effective global batch size of 48. For each input, we sample a group of candidate graphs in Phase-1 and further sample multiple candidate reasoning rollouts for each graph in Phase-2. The generation temperature is set to 0.7, while the maximum prompt length and completion length are set to 4096 and 1024, respectively. For Phase-1 reward construction, the final reward consists of a weighted combination of structural formatting rewards and downstream reasoning utility. Specifically, the leak-controlled reasoning score term is assigned a weight of 80%. For policy regularization, the KL penalty coefficient is set to 0.01.

A.2 Training Data Construction

To ensure balanced multimodal reasoning capability across diverse tasks, we construct the training set from three major benchmark categories: Physical Perception, Diagram and Quantitative Understanding, and Comprehensive Reasoning Evaluation. Specifically, the Physical Perception category includes VSR, TallyQA, and GQA; the Diagram and Quantitative category includes ChartQA, MathVista, and MathVision; and the Comprehensive Reasoning category includes MME. We carefully

control the data distribution to prevent dataset dominance. For the Physical Perception and Comprehensive Reasoning categories, we uniformly sample 1,000 instances from each dataset (VSR, TallyQA, GQA, and MME). For the Diagram and Quantitative Understanding category, we sample 1,000 instances from MathVista, 1,500 instances from MathVision, and 1,500 instances from ChartQA. During training, samples are drawn in a category-balanced manner, while also preserving balanced proportions across datasets within each category. This design improves robustness and generalization across heterogeneous multimodal reasoning tasks.

B Dataset Details

B.1 VSR

The VSR Liu et al. (2023a) benchmark is designed to evaluate spatial relation understanding in vision-language models. In this work, we use the VSR subset of the Visual CoT benchmark Shao et al. (2024a) for both training and evaluation, focusing on the Relation Reasoning category. The task requires models to verify whether a spatial relationship, such as "left of," "inside," or "under," correctly describes the objects in an image. Each sample consists of an image, a textual statement, and a binary answer label, along with ground-truth bounding boxes for the referenced objects. These annotations require models to accurately localize entities before performing spatial reasoning.

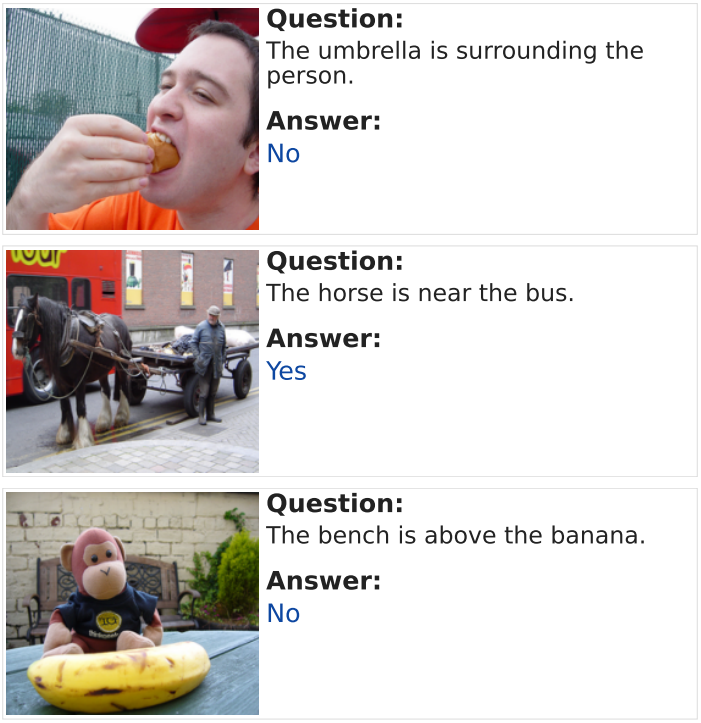


Figure 6: Illustrative examples from the VSR dataset.

The dataset contains diverse real-world scenes with complex object arrangements, providing a challenging benchmark for fine-grained spatial reasoning and visual grounding.

B.2 TallyQA

TallyQA Acharya et al. (2019) is a benchmark for evaluating open-ended counting ability in Visual Question Answering (VQA). Unlike traditional counting datasets that mainly depend on object detection, TallyQA includes both simple and complex counting questions. The complex subset is especially relevant to our work, as it requires models to reason about object relationships, visual attributes, and scene context, such as identifying specific objects under spatial or semantic constraints. Built on images collected from Visual Genome and VQA v2.0, the dataset provides a challenging

testbed for assessing fine-grained visual grounding and advanced numerical reasoning in diverse real-world scenes.

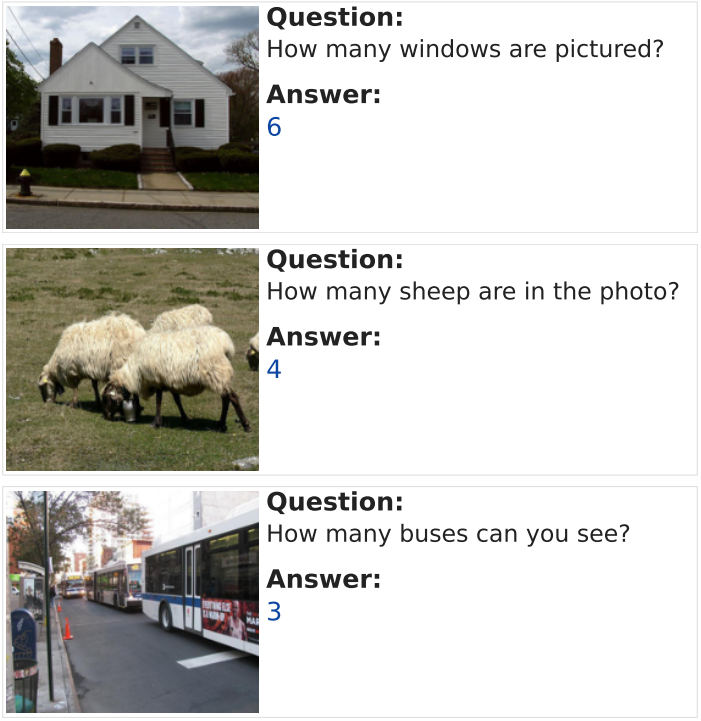


Figure 7: Illustrative examples from the TallyQA dataset.

B.3 GQA

GQA Hudson & Manning (2019) is a large-scale benchmark for visual reasoning and compositional question answering in real-world scenes. Unlike conventional VQA datasets, GQA is built upon the structured scene graphs of Visual Genome, enabling the construction of questions with complex logical dependencies. The benchmark requires models to perform multi-step reasoning, including object localization, attribute recognition, and understanding spatial relationships between objects. By emphasizing compositional reasoning over dataset bias, GQA provides an effective evaluation framework for measuring a model's ability to genuinely interpret and reason about visual scenes.

B.4 ChartQA

ChartQA Masry et al. (2022) is a benchmark for evaluating question answering over charts. It contains around 9.6K human-written questions and 23.1K additional questions derived from chart summaries. Each sample pairs a chart image with a question that requires extracting and reasoning over visual and numerical information. The dataset includes various real-world chart types such as bar, line, and pie charts. Compared to earlier synthetic datasets, ChartQA emphasizes natural questions and supports both direct lookup and multi-step reasoning, making it a challenging benchmark for chart understanding in multimodal models.

B.5 MathVista

MathVista Lu et al. (2023a) is a large-scale benchmark designed to evaluate mathematical reasoning in visual contexts. It consists of 6,141 problems aggregated from 28 existing multimodal datasets, along with three newly introduced datasets: IQTest, FunctionQA, and PaperQA. The benchmark spans a broad range of mathematical topics, including arithmetic, geometry, algebra, and statistics, presented in various visual formats such as charts, diagrams, and geometric figures.



Question:
Is the black car on the left side or on the right?

Answer:
left



Question:
Is there any food to the left of the bowl that is not little?

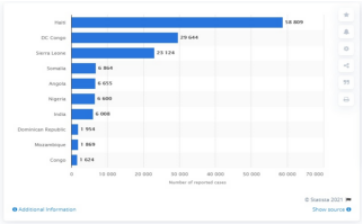
Answer:
yes



Question:
Are there either any traffic lights or clocks?

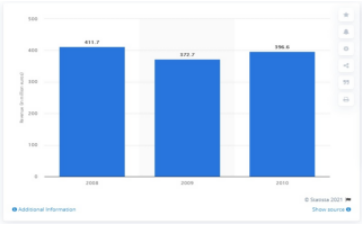
Answer:
no

Figure 8: Illustrative examples from the GQA dataset.



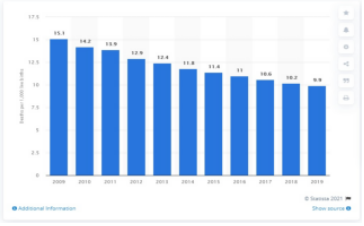
Question:
Which country had the highest reported number of cholera cases?

Answer:
Haiti



Question:
What was the revenue of the Maritim Hotel Group in 2010?

Answer:
396.6

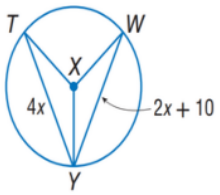


Question:
What was the infant mortality rate in Libya in 2019?

Answer:
9.9

Figure 9: Illustrative examples from the ChartQA dataset.

Unlike standard VQA benchmarks, MathVista requires models to combine precise visual understanding with multi-step reasoning, making it a challenging testbed for modern multimodal large language models.



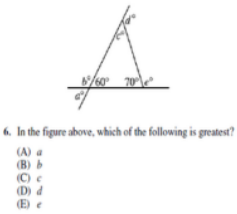
Question:
Arc WTY is congruent to arc TWY. Find the value of x.

Answer:
5



Question:
What does the dial indicate as the top facing number?

Answer:
475



Question:
In the figure above, which of the following is the greatest?

Answer:
d

Figure 10: Illustrative examples from the MathVista dataset.

B.6 MATHVISION

MathVision Wang et al. (2024) is a curated visual mathematical benchmark containing 3,040 high-quality problems collected from real-world math competitions. It covers 16 logical domains, including plane and solid geometry, function transformation, and combinatorics, and is organized into five difficulty levels. Unlike earlier benchmarks, MathVision focuses on combining formal mathematical knowledge with rich visual representations. Each problem requires models to accurately interpret geometric structures or mathematical expressions and perform multi-step reasoning to derive the correct solution, making it a challenging benchmark for evaluating mathematical reasoning in modern VLMs.

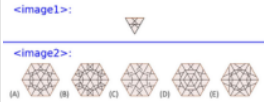
B.7 MME

MME Fu et al. (2025) is a comprehensive benchmark for evaluating the overall capabilities of Multimodal Large Language Models (MLLMs). It includes 14 subtasks organized into two main dimensions: perception and cognition. The perception tasks assess abilities such as object recognition, counting, and spatial awareness, while the cognition tasks focus on higher-level reasoning, including commonsense understanding, numerical computation, and code recognition. A key characteristic of MME is that all question–answer pairs are manually constructed rather than drawn from existing datasets, which helps avoid data contamination and provides a more reliable evaluation of model generalization ability.



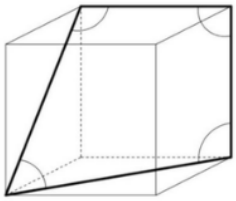
Question:
Alice has four jigsaw pieces. Which two can be fitted together to form a hexagon?

Answer:
B



Question:
Elvis has 6 triangles with this pattern. Which picture can he make with them?

Answer:
A



Question:
In the diagram we see a cube and four marked angles. How big is the sum of those angles?

Answer:
B

Figure 11: Illustrative examples from the MathVision dataset.



Question:
Is this a photo of Komadome-Hachiman-jinja? Please answer yes or no.

Answer:
No



Question:
Is this a picture of Bagakain Lake? Please answer yes or no.

Answer:
No



Question:
Is this a picture of Jindo Bridge? Please answer yes or no.

Answer:
No

Figure 12: Illustrative examples from the MME dataset.

C PROMPTS

C.1 PROMPT FOR RELATIONAL GRAPH CONSTRUCTION

As illustrated in Figure 13, we design a dedicated prompt to guide the model in extracting question-relevant visual relations and organizing them as relational triples and an adjacency-list graph in JSON format. The prompt specifies node categories, standard predicates, and strict output constraints, and is applied consistently during training and evaluation, with outputs terminated by a dedicated </graph> token.

Prompt for Relational Graph Construction

```
You are a structured information extractor for Visual Question Answering
↪  (VQA). Given an image and a question, your task is to extract
↪  information relevant to the question and represent it as "triples +
↪  a graph", so that a downstream model can answer the question using
↪  only the graph and the question (as if it had seen the image).

---------------------------------
[General Requirements]
---------------------------------
- Extract only information that is directly relevant to the question; do
↪  not include irrelevant content
- Do not hallucinate objects or relations that are not present in the
↪  image
- Do NOT provide the final answer
- The output must be strictly valid JSON (no extra text or explanations)

---------------------------------
[Node Requirements]
---------------------------------
Nodes must belong to the following categories (examples are
↪  illustrative, not restrictive):

[Entity] real-world objects or people, e.g., man, woman, dog, car,
↪  bicycle, tree
[Geometric] geometric elements, e.g., point_A, line_1, angle_ABC,
↪  triangle_1
[Chart] chart elements, e.g., bar_2019, line_sales, point_Q1, category_A
[Numeric] numerical values, e.g., value_10, value_50%
[Attribute] descriptive properties, e.g., red, big, small

All nodes must have unique IDs in the format:
type[nodeX], e.g.,
man[node1], bicycle[node2], red[node3]

---------------------------------
[Relation (Edge / Predicate) Requirements]
---------------------------------
Prefer using the following standard predicate set:

[Attribute]: is_a, has_color, has_shape, has_value, has_label
[Spatial]: left_of, right_of, above, below, inside, on, under, near
[Interaction]: holding, riding, wearing, using, connected_to
[Quantitative]: greater_than, less_than, equal_to, approximately,
↪  rank_higher
[Geometric/Structural]: parallel_to, perpendicular_to, intersects,
↪  forms_angle_with, has_length, has_angle, part_of, increasing,
↪  decreasing

You may use semantically equivalent natural expressions (e.g., next_to,
↪  beside, sitting_on), but they must:
- Be semantically mappable to the standard predicates
```

```
- Not be vague or meaningless
- Prefer standard predicates when possible

--------------------------------
[Step 1: Generate Triples]
--------------------------------
Extract a set of triples (subject, predicate, object)

Requirements:
- Include only information necessary to answer the question
- Triples must support a reasoning path from the question entities to
↪  potential answer entities
- All important attributes (e.g., color, size, numerical value) must be
↪  explicitly represented
- Do not include duplicate or conflicting triples
- If a relation is uncertain from the image, do not include it

--------------------------------
[Step 2: Build Graph]
--------------------------------
Construct an adjacency list graph from the triples

Strict constraints:
- The graph must be fully derived from the triples
- Do not add relations not present in the triples
- Do not omit any relations from the triples
- Node names and IDs must exactly match those in the triples
- The graph must contain a complete reasoning path (no missing key
↪  links)
- Do not leave important nodes disconnected

--------------------------------
[Forbidden Behaviors]
--------------------------------
- Do NOT directly output the answer node (e.g., only "dog[nodeX]")
- Do NOT create shortcut structures that bypass reasoning
- Do NOT introduce objects or relations not grounded in the image

--------------------------------
[Output Format (Strict JSON)]
--------------------------------
{
  "triples": [
    ["subject[nodeX]", "predicate", "object[nodeY]"],
    ...
  ],
  "graph": {
    "subject[nodeX]": [
      {"predicate": "predicate", "object": "object[nodeY]"},
      ...
    ],
    ...
  }
}
```

Figure 13: The template instructs the model to extract question-relevant visual information as relational triples and an adjacency-list graph in strict JSON format. It specifies node categories, standard predicates, and output constraints so that downstream reasoning can proceed from structured visual relations rather than isolated entities.

C.2 Prompt for Graph-Centric Stepwise Reasoning

As illustrated in Figure 14, given the constructed graph and the question, the same vision-language model receives a single user message formed by concatenating the instruction below, the ques-

tion, the serialized graph, and a short anti-leak reminder (to avoid relying on hidden text from the graph-construction turn). Each reasoning step uses one of three operations—*Select*, *Explore*, and *Infer*—implemented as `[Find]`, `[Follow]`, and `[Verify]` in the template. The final answer is terminated by a dedicated `</answer>` token.

Prompt for Graph-Centric Stepwise Reasoning

```
You are a graph-centric reasoning assistant. Given a Graph and a
↪  Question, you must reason step by step and produce the answer.

Use the graph as your primary source of information. Only consult the
↪  image if the graph is insufficient, but reasoning should rely on
↪  nodes and relations in the graph as much as possible.

--------------------------------
[Allowed Operations]
--------------------------------

You can ONLY use these three operations:

1. [Find]
Find nodes in the graph that satisfy certain conditions.

- Use graph nodes or predicates as conditions
- Must explicitly mention node IDs or predicates

2. [Follow]
From a given node, follow a predicate to reach another node.

- Must specify the starting node (node ID)
- Must specify the predicate
- The target node must exist in the graph

3. [Verify]
Process or confirm results, such as:
- Determine the type of a node
- Check whether a condition is satisfied
- Produce the final answer

--------------------------------
[Reasoning Rules]
--------------------------------

- Reason step by step
- Use as many steps as needed to complete the reasoning. Do not stop
↪  early if the reasoning is incomplete
- Each step should make progress toward answering the question
- Each step must use exactly ONE operation: [Find], [Follow], or
↪  [Verify]
- Each step must explicitly include node IDs (e.g., woman[node1])
- Do NOT invent nodes or relations not in the graph
- Multi-hop [Follow] is allowed
- All three operations must appear at least once

--------------------------------
[Output Format: JSON only]
--------------------------------

{
  "reasoning": [
    "Step 1: [Find] ... (must include node IDs)",
    "Step 2: [Follow] ... (must include node IDs and predicate)",
    "Step 3: [Verify] ...",
    ...
```

```
  ],
  "answer": "final answer (concise)"
}

Question: {question}
Graph_JSON: {graph_json}
Do not rely on any hidden previous assistant text. Use only Graph_JSON
↪  and optional image.
```

Figure 14: The template guides the model to perform stepwise graph-centric reasoning over the constructed relational graph. Each step uses *Select*, *Explore*, or *Infer*, implemented as `[Find]`, `[Follow]`, and `[Verify]`, and records explicit node references in JSON-formatted trajectories leading to the model's predicted response.

C.3 Prompt for Scoring

To obtain a robust and objective assessment during training, we use GPT-4o as an automated evaluator to score the model's predicted answer against the ground-truth label. The detailed scoring template and criteria are illustrated in Figure 15.

Prompt for Scoring

```
You are responsible for proofreading the answers, you need to give a
↪  score to the model's answer by referring to the standard answer,
↪  based on the given question. The full score is 1 point and the
↪  minimum score is 0 points. Please output the score in the json form
↪  "{score: <score>}". The evaluation criteria require that the closer
↪  the model's answer is to the standard answer, the higher the score.
Question: {question}
Standard answer: {gold_answer}
Model's answer: {predicted_answer}
```

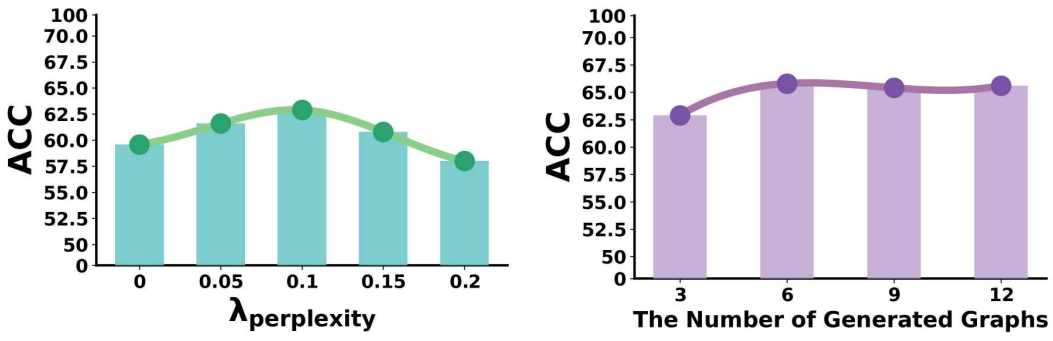
Figure 15: The scoring prompt for GPT-4o-based evaluation. Following the protocol in GRIT Fan et al. (2025), this prompt instructs GPT-4o to act as an impartial judge. It evaluates the alignment between the model's prediction and the ground truth, assigning a score from 0.0 to 1.0 based on the closeness of the predicted answer to the standard answer.

D Sensitivity Analysis

We analyze two key hyperparameters on VSR, including the perplexity-based reward weight and the number of generated graphs.

For the perplexity weight, we observe a clear non-monotonic trend in Figure 16a. A moderate weight consistently improves performance, indicating that uncertainty-aware signals act as an effective regularizer during optimization. Specifically, perplexity reflects the stability of reasoning trajectories under a given graph, suppressing spurious or fragile reasoning paths and encouraging more consistent and robust graph-driven reasoning. However, overly large weights shift optimization toward uncertainty minimization, weakening supervision from answer correctness and degrading performance. Overall, performance remains stable across a wide range, with a moderate weight achieving the best trade-off between stability and accuracy.

For the number of generated graphs, Figure 16b shows that performance first improves and then gradually saturates as more graph candidates are introduced. Generating multiple relational graphs increases structural diversity during exploration, allowing the model to discover more informative visual relations and reducing sensitivity to suboptimal graph construction. However, excessively large graph groups bring limited additional gains, since many generated graphs become structurally similar and contribute redundant supervision signals. This suggests that a moderate number of graph candidates provides the best balance between exploration diversity and optimization efficiency.



(a) (b)

Figure 16: Sensitivity analysis on VSR. (a) $\lambda_{\text{perplexity}}$: moderate values achieve the best stability–accuracy trade-off. (b) Number of generated graphs: performance improves and then plateaus, suggesting diminishing returns beyond a moderate candidate group size.

Model Variant	TallyQA	ChartQA	MathVista
Qwen3-VL-4B (GRATS)	**75.0**	**77.9**	**58.8**
w/o Graph Construction	66.3	72.3	50.5
w/o Graph Reasoning	69.5	73.6	55.0

Table 2: Stage-wise ablation results of GRATS based on Qwen3-VL-4B on representative test sets. The w/o Graph Construction variant removes relational graph construction, while the w/o Graph Reasoning variant removes graph-centric reasoning.

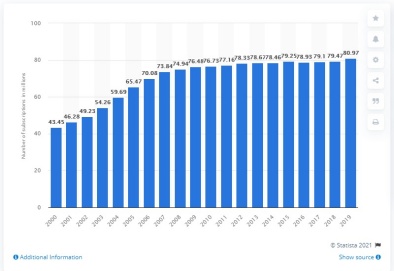
E Stage-wise Ablation Results

To further analyze the contribution of each stage in GRATS, we conduct stage-wise ablation studies based on the Qwen3-VL-4B backbone. Specifically, we evaluate two variants: (1) **w/o Graph Construction**, which removes the relational graph construction stage and directly performs reasoning over visual inputs; and (2) **w/o Graph Reasoning**, which removes graph-centric reasoning while retaining graph construction. The complete results on representative downstream benchmarks are reported in Table 2.

The results demonstrate that both graph construction and graph-centric reasoning are essential for effective multimodal reasoning. Removing graph construction causes larger performance degradation, highlighting the importance of explicit relational visual representations. Meanwhile, removing graph reasoning also consistently decreases performance, indicating that constructing structured graphs alone is insufficient without dedicated reasoning over the learned relational structures.

F More Examples

Case Study 1



Question: How many mobile cellular subscriptions were registered in the UK between 2000 and 2019?

Phase 1: Graph Construction.

- `bar[node1] –has_value–> 43.4`

- `bar[node2] –has_value–> 46.28`
- `bar[node3] –has_value–> 49.21`
- `bar[node4] –has_value–> 54.26`
- `bar[node20] –has_value–> 79.47`
- `bar[node21] –has_value–> 80.97`

(showing 6 of 21 triples for brevity.)

Phase 2: Graph-Centric Reasoning.

1. Step 1: [Find] bar[node21] to identify the value for the year 2019.
2. Step 2: [Follow] bar[node21] using the predicate 'right_of' to confirm its position in the sequence.
3. Step 3: [Verify] The value associated with bar[node21] is 80.97, which represents the number of mobile cellular subscriptions registered in the UK in 2019.

Answer: 80.97 ← `Correct`

Figure 17: During graph construction, the model extracts bar-chart entities and their associated values as relational triples. In the reasoning phase, it performs stepwise graph operations to retrieve the target year, traverse temporal structure, and verify the quantitative answer, ultimately producing the final subscription count.

Case Study 2



Question: Do you see any vending machines or notepads?

Phase 1: Graph Construction.

- `bathroom[node1] –connected_to–> bathroom[node2]`

Phase 2: Graph-Centric Reasoning.

1. Step 1: [Find] bathroom[node1]
2. Step 2: [Follow] bathroom[node1] connected_to bathroom[node2]
3. Step 3: [Verify] No vending machines or notepads are present in the graph or image.

Answer: No ← `Correct`

Figure 18: The figure illustrates how the model organizes scene entities into a relational graph and leverages stepwise graph reasoning to determine object presence, resolving whether target categories appear by consulting both the graph structure and visual context.

Case Study 3



Question: Which food has the least carbs?

Phase 1: Graph Construction.

- `plate[node2] –right_of–> plate[node1]`

- `plate[node2] –under–> bowl[node1]`

Phase 2: Graph-Centric Reasoning.

1. Step 1: [Find] plate[node2] to identify the plate with food items.
2. Step 2: [Follow] plate[node2] to bowl[node1] to understand spatial relationship.
3. Step 3: [Verify] bowl[node1] contains soup, which typically has fewer carbs than sandwiches or pastries.

Answer: Soup ← `Correct`

Figure 19: During graph construction, the model grounds the plates, bowl, and their spatial relations in the image. The final comparison additionally requires nutritional knowledge acquired during pretraining because relative carbohydrate content is not directly observable from pixels. This task-required prior complements rather than replaces visual evidence: the image identifies the candidate foods and their scene relations, while pretrained knowledge supplies their nutritional properties. Therefore, this use of prior knowledge does not constitute drift away from available visual information, and GRATS remains effective in grounding the visual part of the inference.

Failure Case



Question: The bed is facing the tv.

Phase 1: Graph Construction.

- `bed[node1] –right_of–> tv[node2]`

Phase 2: Graph-Centric Reasoning.

1. Step 1: [Find] bed[node1]
2. Step 2: [Follow] bed[node1] right_of tv[node2]
3. Step 3: [Verify] The bed is positioned to the right of the TV, indicating it is not facing the TV.

Answer: False ← `Correct`

Figure 20: Failure case illustrating a graph-reasoning error. The graph correctly records that `bed[node1]` is to the right of `tv[node2]`, but the `[Verify]` step incorrectly treats this spatial relation as evidence that the bed is not facing the TV. Because `right_of` does not determine orientation, the error arises in graph-centric inference rather than graph construction; the final answer is correct only coincidentally. This failure case highlights that graph-centric reasoning enables more fine-grained error attribution. Although the constructed graph correctly captures the visual relation, the model incorrectly infers additional semantic meaning from the predicate `right_of`, indicating a reasoning error rather than a perception failure. Such cases suggest that future improvements should focus on enhancing relation-aware inference, such as incorporating constraints over predicate semantics to prevent invalid logical deductions. Nevertheless, explicitly separating graph construction and reasoning provides a clearer pathway for diagnosing and improving multimodal reasoning systems.