Graph-to-Vision: Multi-graph Understanding and Reasoning using Vision-Language Models

Anonymous ACL submission

Abstract

Recent advances in Vision-Language Models (VLMs) have shown promising capabilities in interpreting visualized graph data, offering a new perspective for graph-structured reasoning beyond traditional Graph Neural Networks (GNNs). However, existing studies focus primarily on single-graph reasoning, leaving the critical challenge of multi-graph joint reasoning underexplored. In this work, we introduce the first comprehensive benchmark designed to evaluate and enhance the multi-graph reasoning abilities of VLMs. Our benchmark covers four common graph types—knowledge graphs, flowcharts, mind maps, and route maps—and supports both homogeneous and heterogeneous graph groupings with tasks of increasing complexity. We evaluate several state-of-the-art VLMs under a multi-dimensional scoring framework that assesses graph parsing, reasoning consistency, and instruction-following accuracy. Additionally, we fine-tune multiple open-source models and observe consistent improvements, confirming the effectiveness of our dataset. This work provides a principled step toward advancing multi-graph understanding and reveals new opportunities for cross-modal graph intelligence.

1 Introduction

Graphs are fundamental for modeling complex relationships and are widely used in domains such as knowledge representation, social networks, and recommendation systems (Wu et al., 2022). With the rise of deep learning, there is growing interest in reasoning over multiple graphs to support tasks like knowledge integration and complex decision-making.

While Graph Neural Networks (GNNs) have shown strong performance in various graph-based tasks (Zhou et al., 2021), they face notable challenges in multi-graph settings—particularly with heterogeneous graph structures—due to scalability limitations and poor generalization (Wu et al., 2023).

In parallel, Vision-Language Models (VLMs) (Chen et al., 2020), which combine Transformer-based encoders for text and images, have demonstrated promising cross-modal reasoning abilities. Recent work suggests that rendering graphs as images and feeding them into VLMs allows better generalization across diverse structures (Zou et al., 2024).Building on this, DeepSeek-OCR(Wei et al., 2025)introduces an LLM-centric paradigm for Contexts Optical Compression. By compressing dense optical information into a minimal set of visual tokens, this approach significantly enhances the efficiency and accuracy of high-resolution document understanding and complex layout interpretation.

However, most existing studies focus on single-graph reasoning. The ability to jointly interpret and reason across multiple graphs—critical for tasks like multi-source alignment or integrative analysis—remains underexplored. To address this, we introduce the first benchmark designed specifically for multi-graph reasoning with VLMs. It covers four common graph types (flowcharts, knowledge graphs, mind maps, and route maps) and includes both homogeneous and heterogeneous groupings with progressively difficult tasks.

We propose a multi-dimensional evaluation framework assessing graph parsing, instruction-following, and reasoning consistency. Using this benchmark, we evaluate several state-of-the-art VLMs and fine-tune open-source models, observing consistent improvements in reasoning capabilities. Despite these contributions, our fine-tuning is currently limited to lightweight models due to the high computational cost of large-scale VLMs, restricting scalability analysis and broader applicability. Our main contributions are as follows:

1. We introduce the first comprehensive benchmark for evaluating and improving the multi-

graph reasoning abilities of VLMs.

2. We systematically evaluate several state-of-the-art VLMs on our benchmark using a dedicated multi-dimensional framework designed for multi-graph reasoning.

3. We fine-tune multiple open-source VLMs on our benchmark and observe consistent improvements in their multi-graph reasoning performance.

2 Related Work

Recent work has increasingly explored Vision-Language Models (VLMs) for graph reasoning, especially through visual modalities. Image-based benchmarks such as GRAPHTMI (Das et al., 2023), VisionGraph (Li et al., 2024), and VGBench (Zou et al., 2024) demonstrate that visual formats often outperform text for structured reasoning. Diagram-oriented datasets like NovaChart (Hu et al., 2024) and DiagramQG (Zhang et al., 2024)further extend this direction to broader reasoning tasks. Despite these advances, recent studies on charts—a structured form of visual graphs—highlight that VLMs remain sensitive to visual perturbations and struggle with complex reasoning (Mukhopadhyay et al., 2024).

To address such limitations, structured visual priors have been incorporated. GITA (Wei et al., 2024) leverages layout-aware visual graphs, and LLaVA-SG (Wang et al., 2025) introduces scene graph intermediates with message passing for relation-aware parsing. In optimization domains, Bridging (Zhao et al., 2025) exploit graph-based visual cues for improved performance without parameter tuning.

While progress in single-graph reasoning is significant, multi-graph joint reasoning remains largely unaddressed. Existing benchmarks lack mechanisms for evaluating cross-graph integration. Our work fills this gap by introducing a dedicated benchmark for multi-graph reasoning and evaluating modern VLMs under a multi-dimensional framework.

3 Dataset

3.1 Overview

We introduce a benchmark specifically designed to evaluate the multi-graph joint reasoning capabilities of Vision-Language Models (VLMs). As illustrated in Figure 1 (a), the benchmark includes

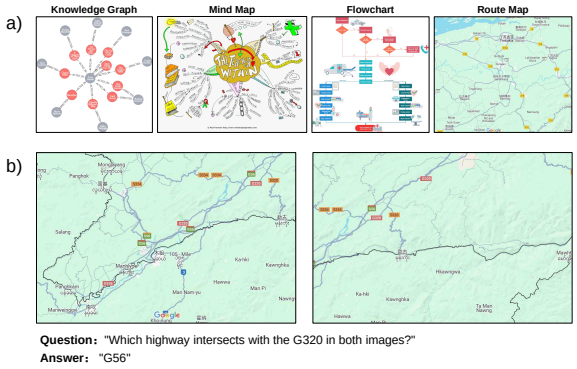


Figure 1: (a) Examples of the four types of graphs included in our benchmark: knowledge graphs, mind maps, flowcharts, and route maps. (b) An example sample from our benchmark, consisting of a set of related graphs, a corresponding instruction, and its reference answer.

four types of graph images—flowcharts, knowledge graphs, mind maps, and route maps—which reflect common structures in real-world reasoning tasks. Each data sample in our benchmark consists of a set of interrelated graph images (e.g., graphs with shared themes, overlapping nodes, or logically connected content), a natural language instruction, and a ground-truth response. An example of such a sample is provided in Figure 1 (b).

To facilitate systematic evaluation, the image sets are organized into two categories: (1) Homogeneous-type groups, where all graphs belong to the same category, and (2) Heterogeneous-type groups, where graphs span different categories. Each instruction is crafted to require reasoning across multiple graphs in the set, thereby assessing a model's ability to jointly interpret and integrate graph-structured information. All instruction-response pairs are initially generated by GPT-4o (OpenAI, 2024b), followed by rigorous human verification, filtering, and refinement to ensure quality, clarity, and consistency.

The remainder of this section is organized as follows: Section 3.2 details the image collection process, including the selection and preprocessing of graph images across different types. Section 3.4 describes how these images are grouped into semantically or structurally related sets to support multi-graph reasoning. Section 3.5 presents our approach for generating instruction-response pairs using GPT-4o, tailored to promote cross-graph comprehension. Finally, Section 3.6 presents a comprehensive statistical analysis of the benchmark, highlighting key characteristics and insights rele-

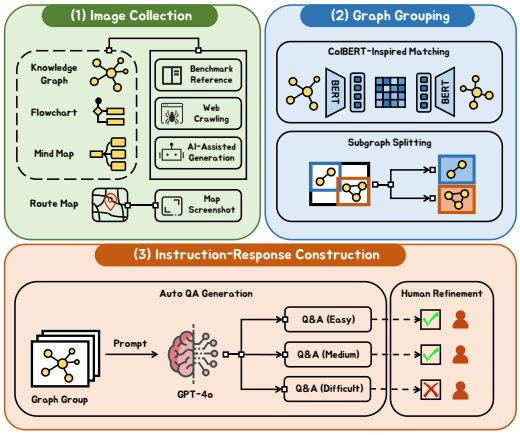


Figure 2: Overview of the benchmark construction pipeline. The process includes: (1) collecting diverse graph images across four types; (2) grouping them into semantically or structurally coherent sets using ColBERT-inspired matching or subgraph splitting; and (3) generating instruction-response pairs via GPT-4o, followed by manual review and refinement to ensure clarity and reasoning quality.

vant to model evaluation. The full data construction pipeline is illustrated in Figure 2.

3.2 Graph Image Collection Strategies

We collected four types of graph images: knowledge graphs, flowcharts, mind maps, and route maps. To ensure both diversity and quality, we employed a combination of data collection strategies, detailed as follows:

Benchmark Referencing and Web Crawling. We first obtained a large number of graph images by referencing the multimodal instruction-following benchmark (Ai et al., 2024), and further expanded the dataset by crawling web images using the benchmark's associated keywords. The keywords were generated by GPT-4o through extracting all nodes and edges from the benchmark graphs and summarizing them. Benchmark referencing ensures high-quality and domain-relevant data, while web crawling enhances diversity by introducing a broader set of publicly available visual formats. These two methods are applicable to the first three graph types.

AI-Assisted Graph Generation. To further enrich knowledge graphs, flowcharts, and mind maps, we adopted an AI-assisted generation approach. The full procedure, including prompt construction, keyword selection, and diagram generation, is detailed in below.

3.3 AI-Assisted Graph Generation

Specifically, we prompted GPT-4o to generate 200 diverse keywords spanning a wide range of domains. The resulting set of domain-specific keywords is illustrated in Figure 3. For each keyword,

Figure 3: The 200 diverse domain-specific keywords generated by GPT-4o, which can be combined with the prompt in Figure 4 to effectively guide GPT-4o in generating high-quality graph descriptions.

GPT-4o was guided to produce a detailed graph description, including node names, edge labels, and connectivity information. The prompt used to guide this process is shown in Figure 4. These

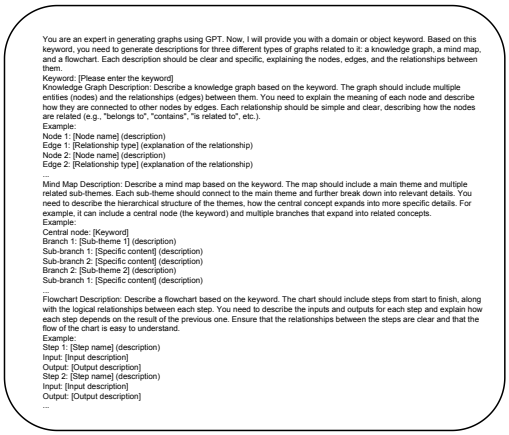


Figure 4: Prompt for guiding GPT-4o to separately generate descriptions of a knowledge graph, a mind map, and a flowchart based on a given keyword.

descriptions were input into the yEd Live drawing tool,[1] which then generated the correspond-

[1]yEd Live is an online graph editor supporting automatic layout for structured graphs. (https://www.yworks.com/y

ing graph images automatically. For example, to generate a mind map on "Machine Learning", we prompted GPT-4o to describe the graph, and used yEd Live's "Create a diagram from text with ChatGPT" to produce the final diagram (see Figure 5). Compared to directly selecting graphs from publicly available datasets, this method offers significant advantages: Public datasets are often limited in scale, domain diversity, or relationship complexity. In contrast, AI-generated graphs can flexibly cover a wider range of topics and structures. By leveraging carefully designed generation strategies, we achieved greater domain coverage, structural diversity, and complexity—enhancing the benchmark's generality and its effectiveness in evaluating models' cross-domain reasoning capabilities.

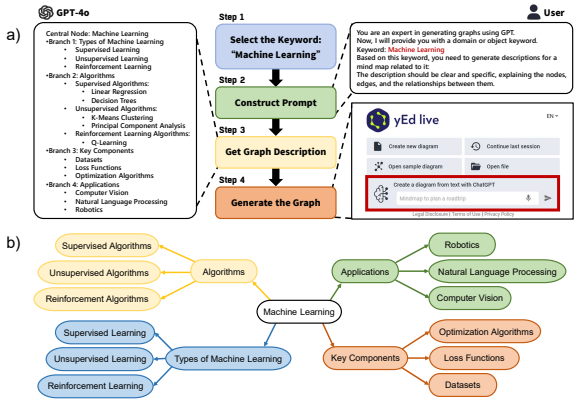


Figure 5: (a) The system pipeline for generating a mind map using the keyword "Machine Learning." The process starts with keyword selection, followed by prompt construction, graph description generation, and automated mind map rendering using the yEd Live "Create a diagram from text with ChatGPT" tool. (b) An example of the generated mind map corresponding to the keyword.

High-Confidence Route Maps via Google Maps. For route maps, we adopted a targeted strategy of capturing high-resolution screenshots from Google Maps. This ensured the geographic accuracy and visual clarity of the maps, supporting more reliable downstream visual reasoning and inference.

3.4 Grouping Graph Images by Semantic and Structural Relevance

To construct semantically coherent image groups, we employed tailored grouping strategies based on graph type. Specifically, knowledge graphs, flowcharts, and mind maps—due to their conceptual overlap and structural compatibility—were grouped both within the same type and across different types. In contrast, route maps, which primarily convey spatial and navigational information, differ fundamentally from the other categories. As such, they were grouped only within their own type to preserve thematic consistency and interpretability.

3.4.1 ColBERT-Inspired Graph Grouping Strategy

For the first three types of graphs, we adopted a **C**olBERT-**I**nspired **G**raph **G**rouping (**CIGG**) strategy, which leverages fine-grained token-level similarity to construct semantically meaningful image groups:

1. **Graph Element Extraction:** We first designed prompts to leverage GPT-4o for extracting all node and edge names from each graph (see Appendix A.1 for prompt details).

2. **Semantic Encoding:** Each extracted node and edge name was encoded into semantic vectors using BERT, where the final-layer hidden states serve as the token-level representations.

3. **ColBERT-Inspired Similarity Matching:** We adopt a bi-directional max-sim approach inspired by ColBERT ((Khattab and Zaharia, 2020)) to compute graph similarity, enabling the construction of semantically coherent graph groups. See Appendix A.2 for details.

3.4.2 Subgraph-Splitting Strategy for Route Maps

Due to the high structural and semantic homogeneity among route maps, distinguishing them in the BERT semantic space is challenging. Thus, the CIGG strategy is not suitable for this graph type. To overcome this, we apply a subgraph-splitting strategy, detailed in Appendix A.3.

3.4.3 Manual Refinement

Each constructed image group was manually reviewed, and those lacking meaningful semantic connections among the graphs were directly discarded. This step ensures that the remaining groups consist of graphs that are conceptually related and suitable for joint reasoning.

ed-live/)

3.5 Instruction-Response Construction

3.5.1 VLM-based Instruction-Response Candidate Generation

We carefully designed prompts to guide GPT-4o in generating effective instruction-response pairs for each group of graph images (see Appendix A.4 for prompt details). For every graph group, GPT-4o is prompted to produce three instruction-response pairs with increasing difficulty levels: easy, medium, and difficult. Each pair is required to involve reasoning across as many graphs in the group as possible, ensuring that the task truly reflects the challenge of multi-graph joint understanding.

3.5.2 Manual Review and Refinement of Instruction-Response Pairs

As a benchmark, the quality of samples is critical. After initial generation, we conducted a rigorous manual review process to ensure each instruction-response pair met quality standards. The full set of review criteria and editing actions are provided in Appendix B.

3.6 Data Statistics and Analysis

To assess the quality of our data, we randomly selected 10% of the samples and invited independent annotators who were not involved in benchmark construction to evaluate the validity of the instruction-response pairs. All reviewed samples were deemed valid, further confirming the overall reliability of our benchmark. In addition, we partitioned the dataset into training, validation, and test splits. Importantly, the test set was carefully curated to ensure comprehensive coverage of different graph group types and difficulty levels, enabling robust and balanced evaluation of model performance. An overview of the dataset composition is provided in Table 1. Additional benchmark statistics are provided in Appendix C.

4 Evaluation of Large Vision-Language Models on Multi-Graph Joint Reasoning

In this section, we present a comprehensive and systematic evaluation of several state-of-the-art VLMs on the proposed benchmark.

The evaluated models span both proprietary and open-source systems, including GPT-4o-mini (OpenAI, 2024a)(OpenAI), Gemini-1.5-pro (DeepMind, 2024) (Google DeepMind), QVQ-72B-Preview[2] (Team, 2024), Qwen2.5-VL-32B-Instruct[3], and Qwen2.5-VL-72B-Instruct[4] (Bai et al., 2025).

4.1 What Abilities Do We Focus On?

Unlike conventional VQA, multi-graph joint reasoning demands a richer evaluation protocol. We propose three dimensions to reflect its structural, semantic, and procedural complexity.

Graph Parsing Accuracy (GPA). This dimension evaluates the model's ability to *comprehend and interpret the structural features of graphs* and *effectively apply this understanding within the context of the question*. Accurate graph parsing is critical for successful multi-graph reasoning.

Reasoning Consistency and Completeness (RCC). This dimension measures the logical consistency and completeness of the model's reasoning process. It reflects whether the model's response demonstrates a *coherent, well-structured, and internally consistent reasoning chain*.

Instructional Reasoning Accuracy (IRA). This dimension assesses whether the model can accurately follow the given instructions to *generate correct or plausible answers*. It directly reflects the model's fundamental capacity for instruction-driven reasoning.

4.2 Evaluation Strategy

In this section, we outline the evaluation strategy employed to assess model performance on the test set derived from our benchmark, as described in Section 3.6. The test set consists of 300 samples, which encompass a diverse range of graph group types and reasoning difficulty levels.

We adopt a two-stage, GPT-assisted evaluation strategy. In the first stage, GPT-4o is tasked with evaluating model responses across the three dimensions defined in Section 4.1. A dedicated prompt is designed for GPT-4o to assess each dimension on a 5-point scale (1-5), where higher scores indicate better performance and stronger capability in the respective criterion. The detailed evaluation prompt used for the assessment process is provided .

The evaluation prompt used to guide GPT-4o in assessing model responses across three reasoning

[2] https://huggingface.co/Qwen/QVQ-72B-Preview

[3] https://huggingface.co/Qwen/Qwen2.5-VL-32B-Instruct

[4] https://huggingface.co/Qwen/Qwen2.5-VL-72B-Instruct

Table 1: An overview of our multi-graph joint reasoning benchmark.

	# Train	# Valid	# Test	# Overall
Knowledge Graph-type	466	72	35	573
Flowchart-type	465	64	57	586
Route Map-type	444	58	62	564
Mind Map-type	475	82	42	599
Heterogeneous-type	919	114	104	1137
Overall	2769	390	300	3459

```
You are an expert evaluator assessing the performance of a multimodal AI model on a multi-graph
reasoning benchmark.
Given the following question, the model's answer, and the human-written reference answer, please
rate the model's response along the following three evaluation dimensions. Each score must be an
integer from 1 to 5, where 1 indicates very poor performance and 5 indicates excellent performance.
---------------------
Question: {question}
Model's answer: {model_ans}
Reference answer: {standard_ans}
---------------------
Evaluation Criteria:
1. Graph Parsing Accuracy (GPA)
   Evaluate whether the model accurately identifies and interprets key structural features of the
involved graphs (such as node relationships, edge directions, clusters, paths, etc.) and appropriately
integrates this structural information into its answer.
   - Score higher if the model demonstrates awareness of graph-specific elements.
   - Score lower if it ignores, misreads, or misrepresents structural relationships.
2. Instructional Reasoning Accuracy (IRA)
   Evaluate whether the model correctly follows the instruction or task implied in the question.
   - Score higher if the model precisely follows the instruction and directly addresses the task.
   - Score lower if the model responds vaguely, omits critical instruction elements, or answers a
different question.
3. Reasoning Consistency and Completeness (RCC)
   Assess the quality of the model' s reasoning process.
   - Score higher if the explanation is coherent, logically ordered, and supports the final answer.
   - Score lower if the reasoning is fragmented, contains contradictions, or lacks key inference steps.

Output format:
Output only three integers separated by a single space (e.g., `4 3 5'). Do not include any explanation,
commentary, or punctuation.

Output example:
3 2 4
```

Figure 6: Prompt used to instruct GPT-4o to evaluate model-generated answers based on three core reasoning criteria: Graph Parsing Accuracy (GPA), Instructional Reasoning Accuracy (IRA), and Reasoning Consistency and Completeness (RCC). The prompt defines each criterion with specific expectations and provides scoring guidance to ensure consistent and fine-grained evaluation.

dimensions is shown in Figure 15.

In the second stage, we randomly select a subset of evaluation samples for human annotation. The human annotators are blinded to the automatic evaluation scores to eliminate bias. Importantly, the same evaluation dimensions and scoring criteria used in the automated evaluation are applied in the human assessment. Once the human evaluation is complete, we compute the correlation between the automatic evaluation scores and human judgments to assess the reliability and validity of the automated evaluation process.

4.3 Evaluation Results and Analysis

In this section, we present a comprehensive analysis of model performance from multiple perspectives.

Overview of Key Results. We summarize overall model performance in Section 4.3.1 and include human-machine consistency analysis in Section 4.3.2. For more detailed analyses across graph group types and difficulty levels, please refer to Appendix D.3.1 and Appendix D.3.2.

4.3.1 Analysis of Overall Model Performance via Automatic Evaluation

Table 2: Average scores assigned by GPT-4o to each model across the three evaluation dimensions: graph parsing accuracy (GPA), reasoning consistency and completeness (RCC), and instructional reasoning accuracy (IRA).

	IRA	GPA	RCC
GPT-4o-mini	3.88	3.73	4.25
Gemini-1.5-pro	3.81	3.78	4.29
QVQ-72B-Preview	3.62	3.78	4.22
Qwen2.5-VL-32B-Instruct	**3.90**	**4.02**	**4.58**
Qwen2.5-VL-72B-Instruct	3.76	3.78	4.21

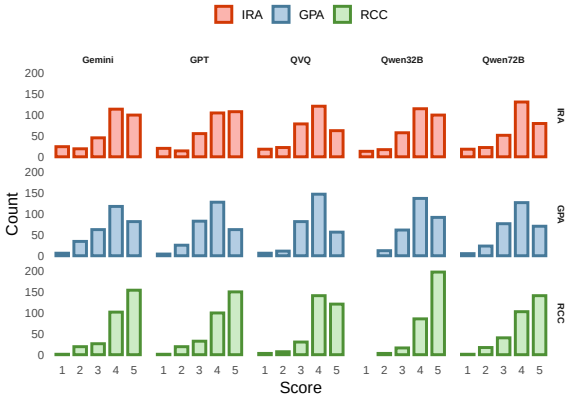


Figure 7: Score distribution histograms across three evaluation dimensions (GPA, RCC and IRA) for each of the five models. Each subplot shows the frequency of scores ranging from 1 to 5, where bars are colored by evaluation dimension.

Table 2 reports the average scores of each model across the three evaluation dimensions, while Figure 7 shows the score distributions per dimension for each model.

We observe that Qwen2.5-VL-32B-Instruct excels in the RCC dimension, suggesting strong semantic organization and abstract reasoning capabilities, which highlights its proficiency in handling cross-node, multi-hop reasoning tasks. Gemini-1.5-pro does not stand out in any single dimension but demonstrates stable and well-balanced performance overall. In contrast, QVQ-72B-Preview exhibits lower average scores and greater variance across dimensions, indicating less consistent performance. Overall, RCC emerges as the highest-scoring and most stable dimension across all models.

4.3.2 Human Evaluation and Consistency Analysis with Automatic Evaluation Scores

To evaluate the consistency between automatic evaluation scores and human judgments, we randomly sampled 10% of the evaluation dataset for manual annotation. The results of the human evaluation are summarized in Table 3, and a visualization of score consistency is provided in Figure 8. We calculated

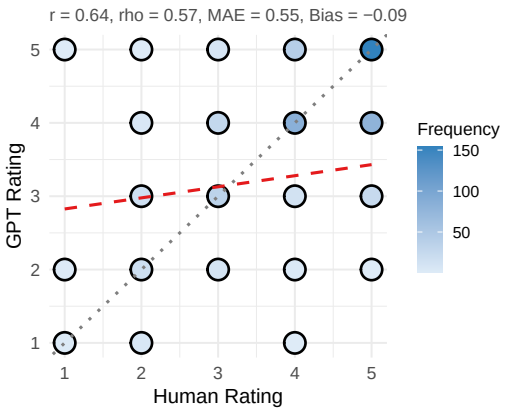


Figure 8: Scatter plot illustrating the consistency between automatic evaluation scores and human ratings across a 10% randomly sampled subset of the evaluation dataset. Each point represents the frequency of a specific score pair (Human, GPT), with color intensity indicating occurrence frequency. The diagonal dotted line denotes perfect agreement, while the red dashed line shows the linear regression trend, representing the alignment between automatic evaluation scores and human scores—closer proximity to the diagonal indicates stronger consistency.

four commonly used consistency metrics: the Pearson correlation coefficient (r), the Spearman rank correlation coefficient (ρ), the Mean Absolute Error (MAE), and Bias. The formal definitions and computation formulas for these metrics are provided in Appendix D.1. The results indicate a moderately strong linear correlation ($r = 0.64$) and a moderate rank correlation ($\rho = 0.57$) between automatic evaluation and human scores, suggesting that the model generally captures overall scoring trends, though some discrepancies remain in the ranking of individual samples. The MAE is 0.55, indicating that the average deviation between model and human scores is less than one point, and the overall error remains within an acceptable range. The bias is -0.09, showing that the model tends to slightly underestimate human scores. To better understand subtle differences in scoring behaviors, especially where discrepancies occurred between humans and GPT-4o, we conducted a detailed dimension-wise analysis, presented in Appendix D.2.

Despite certain imperfections, our dedicatedly designed evaluation prompt enables the automatic evaluation scores to align well with human judgments in terms of overall trends.

5 Fine-Tuning Open-Source Vision-Language Models

Baseline Models. We evaluate six representative lightweight multimodal models as our baselines: DeepSeek-VL-1.3B-Chat[5] (Lu et al., 2024), InternVL2-1B, InternVL2.5-1B, InternVL2.5-1B-MPO[6], Janus-1.3B[7] (Wu et al., 2024), and mPLUG-Owl3-1B-241014[8] (Ye et al., 2024).

Experiment Details. We fine-tuned each model using the training set described in Section 3.6 and evaluated their performance on the same evaluation dataset introduced in Section 4.2, enabling a direct comparison before and after fine-tuning. The evaluation followed the protocol outlined in Section 4.1, assessing model outputs along three dimensions: graph parsing accuracy (GPA), reasoning consistency and completeness (RCC), and instructional reasoning accuracy (IRA). Scores were assigned using GPT-4o as an automatic evaluation. The feasibility and reliability of this automatic evaluation approach were thoroughly validated in the previous section.

5.1 Results

The average scores of each model across the three evaluation dimensions before and after fine-tuning are presented in Table 4, with the corresponding

[5] https://github.com/deepseek-ai/DeepSeek-VL
[6] https://github.com/OpenGVLab/InternVL
[7] https://github.com/deepseek-ai/Janus
[8] https://github.com/X-PLUG/mPLUG-Owl

Table 3: Average scores assigned by human evaluators to each model across the three evaluation dimensions. The scores are based on manual assessment of 10% randomly sampled entries from the evaluation dataset (for comparison, the scores in parentheses denote the automatic evaluation results).

	IRA	GPA	RCC
GPT-4o-mini	4.22(3.88)	3.64(3.73)	4.39(4.25)
Gemini-1.5-pro	4.14(3.81)	3.89(3.78)	4.50(4.29)
QVQ-72B-Preview	4.06(3.62)	3.58(3.78)	4.22(4.22)
Qwen2.5-VL-32B-Instruct	**4.25**(3.90)	**4.00**(4.02)	**4.67**(4.58)
Qwen2.5-VL-72B-Instruct	4.03(3.76)	3.61(3.78)	4.47(4.21)

Table 4: Average scores of each model before and after fine-tuning across the three evaluation dimensions: GPA, RCC and IRA. Fine-tuning generally improves performance across all dimensions.

	IRA		GPA		RCC	
	Before	**After**	**Before**	**After**	**Before**	**After**
DeepSeek-VL-1.3B-Chat	1.77	2.85	1.96	2.70	2.16	2.98
InternVL2-1B	2.38	2.73	2.43	2.64	2.77	2.90
InternVL2.5-1B	2.67	**2.95**	2.57	**2.89**	2.94	**3.10**
InternVL2.5-1B-MPO	**2.69**	**2.95**	**2.72**	2.75	**3.05**	3.06
Janus-1.3B	2.21	2.45	2.31	2.39	2.52	2.65
mPLUG-Owl3-1B-241014	2.13	2.43	2.21	2.46	2.57	2.74

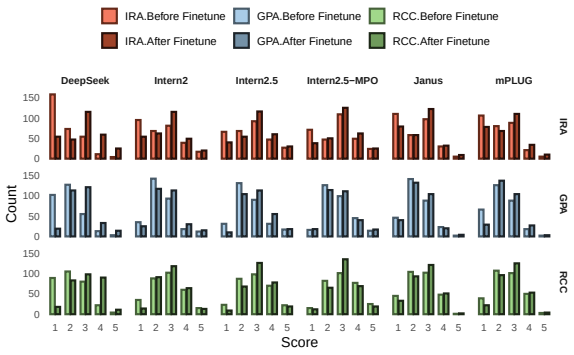


Figure 9: Score distributions for each VLM across three evaluation dimensions: GPA, RCC and IRA. Each row corresponds to a specific evaluation dimension, and each column to a different model. For each score level (1–5), lighter bars indicate results before finetuning, while darker bars represent results after finetuning.

score distributions shown in Figure 9. All models show improvements across all dimensions after fine-tuning.

Notably, DeepSeek-VL-1.3B-Chat achieves the most substantial performance gain, while other models exhibit more modest improvements. This aligns with the observed loss trajectory during fine-tuning: while DeepSeek-VL-1.3B-Chat required nearly 500 iterations to reach its lowest validation loss, most other models began to overfit—evidenced by increasing validation loss—within 200 iterations. These findings suggest that, compared to its counterparts, DeepSeek-VL-1.3B-Chat has stronger generalization capacity in multi-graph joint reasoning tasks and can benefit more from extended fine-tuning.

We also conducted a variance analysis of model scores before and after fine-tuning; detailed results are provided in Appendix D.4.

6 Conclusion

This work investigates multi-graph joint reasoning with VLMs as an alternative to graph neural networks. We propose a benchmark that addresses gaps in data and evaluation protocols. Experiments show that while state-of-the-art VLMs demonstrate strong potential, they also face challenges in structural and semantic integration. Fine-tuning lightweight open-source models on our benchmark yields consistent gains, validating its effectiveness and generalizability.

7 Limitations

Due to the high computational cost of large VLMs, our fine-tuning is limited to smaller models, preventing scalability analysis. Moreover, the benchmark does not yet support non-visual formats like tables, limiting its use in multimodal scenarios involving graph–table combinations. Experiments reveal that while Qwen2.5-VL-32B-Instruct excels in multi-hop reasoning (RCC) and Gemini-1.5-pro provides the most balanced performance, a gap remains in structural integration. Fine-tuning lightweight models like DeepSeek-VL-1.3B yields significant gains, proving the benchmark's value for model alignment. However, we observed that fine-tuning can sometimes impair original structural parsing in complex tasks, such as flowchart logic.

References

Qihang Ai, Jiafan Li, Jincheng Dai, Jianwu Zhou, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2024. Advancement in graph understanding: A multimodal benchmark and fine-tuning of vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7485–7501.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. *Preprint*, arXiv:1909.11740.

Debarati Das, Ishaan Gupta, Jaideep Srivastava, and Dongyeop Kang. 2023. Which modality should i use–text, motif, or image?: Understanding graphs with large language models. *arXiv preprint arXiv:2311.09862*.

Google DeepMind. 2024. Gemini 1.5 pro technical overview. https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024.

Linmei Hu, Duokang Wang, Yiming Pan, Jifan Yu, Yingxia Shao, Chong Feng, and Liqiang Nie. 2024. Novachart: A large-scale dataset towards chart understanding and generation of multimodal large language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3917–3925.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Yunxin Li, Baotian Hu, Haoyuan Shi, Wei Wang, Longyue Wang, and Min Zhang. 2024. Visiongraph: Leveraging large multimodal models for graph theory problems in visual context. *arXiv preprint arXiv:2405.04950*.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Srija Mukhopadhyay, Adnan Qidwai, Aparna Garimella, Pritika Ramu, Vivek Gupta, and Dan Roth. 2024. Unraveling the truth: Do vlms really understand charts? a deep dive into consistency and robustness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16696–16717.

OpenAI. 2024a. Gpt-4o-mini. https://platform.openai.com/docs/models/gpt-4o-mini.

OpenAI. 2024b. Gpt-4o technical overview. https://openai.com/index/gpt-4o.

Qwen Team. 2024. Qvq: To see the world with wisdom.

Jingyi Wang, Jianzhong Ju, Jian Luan, and Zhidong Deng. 2025. Llava-sg: Leveraging scene graphs as visual semantic expression in vision-language models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Haoran Wei, Yaofeng Sun, Yukun Li, and 1 others. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Yanbin Wei, Shuai Fu, Weisen Jiang, Zejian Zhang, Zhixiong Zeng, Qi Wu, James Kwok, and Yu Zhang. 2024. Gita: Graph to visual and textual integration for vision-language graph reasoning. *Advances in Neural Information Processing Systems*, 37:44–72.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. 2024. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *Preprint*, arXiv:2410.13848.

Fang Wu, Siyuan Li, Lirong Wu, Dragomir Radev, and Stan Z. Li. 2023. Discovering and explaining the representation bottleneck of graph neural networks from multi-order interactions. *Preprint*, arXiv:2205.07266.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2024. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. *Preprint*, arXiv:2408.04840.

Xinyu Zhang, Lingling Zhang, Yanrui Wu, Muye Huang, Wenjun Wu, Bo Li, Shaowei Wang, and Jun Liu. 2024. Diagramqg: A dataset for generating concept-focused questions from diagrams. *arXiv preprint arXiv:2411.17771*.

Jie Zhao, Kang Hao Cheong, and Witold Pedrycz. 2025. Bridging visualization and optimization: Multimodal large language models on graph-structured combinatorial optimization. *arXiv preprint arXiv:2501.11968*.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2021. Graph neural networks: A review of methods and applications. *Preprint*, arXiv:1812.08434.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. 2024. Vgbench: Evaluating large language models on vector graphics understanding and generation. *arXiv preprint arXiv:2407.10972*.

A Token Extraction and Similarity Computation for Graph Grouping

A.1 GPT-4o Prompt for Graph Token Extraction

```
You are a graph parser.
I will provide you with an image that contains a graph.
Please carefully examine the image and extract all the nodes and edges present in the graph.
For each node, identify a unique ID or label (as shown in the image).
For each edge, specify the source and target node IDs.
Ignore any non-graphical elements (such as decorative text, titles, or legends).
Only include nodes and edges that are part of the actual graph structure.
Return your result in the following JSON format:
{
  "nodes": [
    {"id": "A"},
    {"id": "B"},
    {"id": "C"}
  ],
  "edges": [
    {"source": "A", "target": "B"},
    {"source": "B", "target": "C"}
  ]
}
```

Figure 10: Prompt used to guide GPT-4o in parsing graph images into structured node and edge representations. The prompt instructs the model to identify all graphical nodes and edges from a given image while ignoring non-structural elements such as decorative text, titles, or legends.

The prompt used to instruct GPT-4o to extract node and edge tokens from individual graph images is shown in Figure 10.

A.2 Bi-directional Max-Sim Similarity Computation

Let G^A and G^B be two graphs. From each graph, we extract the set of node and edge tokens:

$$T^A = \{t_1^A, t_2^A, \ldots, t_{n_A}^A\}, \quad T^B = \{t_1^B, t_2^B, \ldots, t_{n_B}^B\} \tag{1}$$

Each token t_i is encoded into a dense vector $\mathbf{e}_i \in R^d$ using a pretrained BERT encoder. Denote the resulting embeddings as:

$$E^A = \{\mathbf{e}_1^A, \ldots, \mathbf{e}_{n_A}^A\}, \quad E^B = \{\mathbf{e}_1^B, \ldots, \mathbf{e}_{n_B}^B\} \tag{2}$$

We define the *max-sim* score from A to B as:

$$\text{MaxSim}(A \to B) = \frac{1}{n_A} \sum_{i=1}^{n_A} \max_j \cos(\mathbf{e}_i^A, \mathbf{e}_j^B) \tag{3}$$

Similarly, the reverse direction is computed as:

$$\text{MaxSim}(B \to A) = \frac{1}{n_B} \sum_{j=1}^{n_B} \max_i \cos(\mathbf{e}_j^B, \mathbf{e}_i^A) \tag{4}$$

The final *bi-directional similarity score* is the average of both directions:

$$\text{BiMaxSim}(A, B) = \frac{1}{2}[\text{MaxSim}(A \to B) + \text{MaxSim}(B \to A)] \tag{5}$$

We use $\text{BiMaxSim}(A, B)$ as the similarity metric to identify and group semantically aligned graph image sets. This approach captures asymmetric alignment and encourages mutual relevance between token sets from two graphs.

A.3 Subgraph-Splitting Strategy for Route Maps

Due to the high structural and semantic homogeneity among different route maps, we empirically observed that distinguishing them within the BERT semantic space is challenging. Consequently, the CIGG strategy is not suitable for this graph type, as it struggles to reliably identify route maps with semantically related content.

To address this limitation, we adopt a subgraph-splitting strategy with two distinct configurations:

- **Splitting into Two Subgraphs:** We adopt three splitting methods—vertical, horizontal, and diagonal—to divide a route map into two subgraphs (see Figure 11 (a), (b), (c)).
- **Splitting into Four Subgraphs:** We apply a corner-based splitting approach, segmenting the route map into four quadrants to generate four interrelated subgraphs (see Figure 11 (d)).

These methods simulate different spatial partitions while preserving local structural coherence.

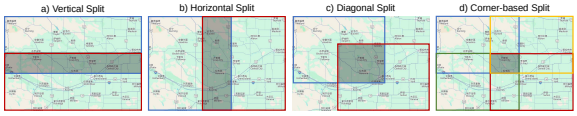


Figure 11: This figure illustrates four distinct methods for splitting route maps into subgraphs to generate image groups. **Vertical Split:** The map is split vertically into two subgraphs, marked by red and blue bounding boxes. **Horizontal Split:** The map is split horizontally into two subgraphs, indicated by red and blue boxes. **Diagonal Split:** The map is split diagonally into two overlapping subgraphs, shown in red and blue. **Corner-based Split:** The map is split into four subgraphs, each represented by yellow, green, blue, and red boxes. The shaded overlapping regions in each method indicate areas where subgraphs share common information, with the overlap controlled at 20%–30% of the original map's area.

A.4 Prompt for Instruction-Response Pair Generation

The full prompt used to guide GPT-4o in generating instruction-response pairs for multi-graph reason-

First, accurately recognize the content in the given multiple related images to ensure a correct understanding of each image. Next, generate three questions based on these images, ensuring they meet the following requirements:
1. Each question must involve content from at least two images. When referring to an image, avoid using sequential indicators such as "the first image" or "the second image." Instead, use descriptive references.
2. The three questions should be labeled with difficulty levels: "easy," "medium," and "difficult."
3. Each question should focus on joint reasoning based on the image content, rather than containing two or more independent subquestions.
4. Provide a complete and logically sound standard answer for each question.
5. The output must strictly follow the JSON format below, ensuring that both questions and answers are in string format:

```
[
    {
        "difficulty": "easy",
        "question": "",
        "answer": ""
    },
    {
        "difficulty": "medium",
        "question": "",
        "answer": ""
    },
    {
        "difficulty": "difficult",
        "question": "",
        "answer": ""
    }
]
```

Make sure to follow all the above requirements and ensure the JSON format is correctly structured.

Figure 12: Prompt used to instruct GPT-4o in generating instruction-response pairs from multiple semantically related graph images.

ing tasks is shown in Figure 12.

B Manual Review and Refinement Procedure

To ensure the quality and consistency of instruction-response pairs, we performed a detailed manual review following the initial generation by GPT-4o. This review process aimed to filter out low-quality samples and correct flawed responses to ensure the benchmark faithfully assesses multi-graph reasoning capabilities.

We identified three main reasons for deeming a sample invalid:

- The instruction only referenced a single graph, violating the benchmark's goal of assessing joint reasoning across multiple graphs.
- The instruction was irrelevant or unrelated to the graph content, resulting in semantically meaningless samples.
- The response contained factual inaccuracies or logical inconsistencies despite the instruction being valid.

Based on the above criteria, we applied the following corrective actions:

- Pairs with invalid instructions were entirely removed from the benchmark.
- Pairs with valid instructions but flawed responses were manually edited to correct errors while preserving the original reasoning intent.

This annotation process contributed significantly to the reliability of the released benchmark.

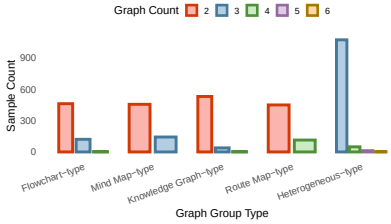


Figure 13: The distribution of samples across five graph group types, categorized by the number of graphs (graph count) within each sample. Each bar represents the number of samples containing a specific number of graphs within each graph group type. Different colors indicate different graph counts.

C Additional Benchmark Statistics

Figure 13 shows the distribution of the number of graph images per sample, where each sample refers to a group of graph images along with its corresponding instruction-response pair. The distribution of test samples by graph group type and difficulty is shown in Figure 14.

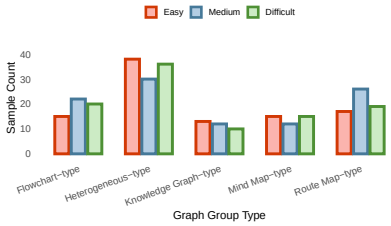


Figure 14: Sample distribution across different graph group types and difficulty levels in the test set. Each bar represents the number of samples for a specific difficulty level (easy, medium, difficult) within each graph group type.The total number of samples for easy, medium, and difficult difficulties are 98, 102, and 100, respectively.

D Prompt for GPT-4o-Based Evaluation

The evaluation prompt used to guide GPT-4o in assessing model responses across three reasoning dimensions is shown in Figure 15.

D.1 Formal Definitions of Consistency Metrics

To quantify the consistency between automatic model scoring and human annotations, we adopt four commonly used statistical metrics: the **Pearson correlation coefficient** (r), the **Spearman rank correlation coefficient** (ρ), the **Mean Absolute Error** (MAE), and **Bias**. Their formal definitions and computation formulas are provided below. Throughout these definitions, we denote x_i as the

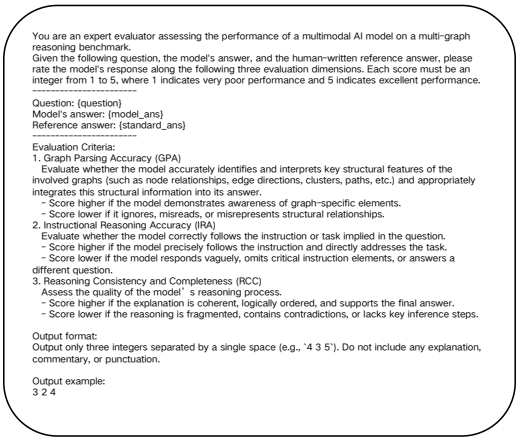


Figure 15: Prompt used to instruct GPT-4o to evaluate model-generated answers based on three core reasoning criteria: Graph Parsing Accuracy (GPA), Instructional Reasoning Accuracy (IRA), and Reasoning Consistency and Completeness (RCC). The prompt defines each criterion with specific expectations and provides scoring guidance to ensure consistent and fine-grained evaluation.

human-assigned score and y_i as the corresponding model-assigned score.

D.1.1 Pearson Correlation Coefficient (r)

The Pearson correlation measures the linear relationship between human and model scores. Given paired scores $\{(x_i, y_i)\}_{i=1}^{n}$, it is defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (6)$$

where $\bar{x}$ and $\bar{y}$ denote the means of x_i (human) and y_i (model), respectively.

D.1.2 Spearman Rank Correlation Coefficient (ρ)

Spearman's ρ measures the monotonic relationship between the rankings of human and model scores. It is computed as:

$$\rho = r(\text{rank}(x), \text{rank}(y)) \quad (7)$$

where $\text{rank}(x_i)$ and $\text{rank}(y_i)$ denote the ranks of the human and model scores, respectively.

D.1.3 Mean Absolute Error (MAE)

MAE evaluates the average magnitude of the absolute differences between model and human scores:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \quad (8)$$

D.1.4 Bias

Bias captures the average signed deviation of model scores from human scores:

$$\text{Bias} = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i) \quad (9)$$

These metrics together provide a comprehensive view of the alignment between automatic evaluation and human judgment.

D.2 Dimension-Wise Bias Analysis between Human and Automatic Evaluation

A further dimension-wise analysis reveals that automatic evaluation scores are generally higher than human scores in the GPA dimension. In contrast, human scores tend to exceed those of the model in the IRA and RCC dimensions. This discrepancy may stem from differing evaluation emphases.

In GPA, GPT-4o often assigns favorable scores when responses include surface-level mentions of nodes, edges, or substructures. Human annotators, however, emphasize deeper structural comprehension—such as hierarchical relationships, edge semantics, and the underlying logical organization of the graph—leading to more conservative assessments.

Conversely, in IRA and RCC, human evaluators are generally more forgiving of minor language inconsistencies, as long as the semantic content is preserved. In contrast, GPT-4o tends to assign lower scores in these cases, likely due to its stricter adherence to pattern-based matching and surface fluency.

This difference underscores the need to consider complementary human and automatic assessments in evaluating VLM performance.

D.3 Additional Evaluation Results

D.3.1 Evaluation Across Graph Group Types

We systematically analyzed the performance of five models across three evaluation dimensions on five graph group types defined in our benchmark: flowchart, knowledge graph, mind map, route map, and heterogeneous. The results are presented in Figure 16. In terms of average scores, model performance across different graph group types largely aligns with the overall trends observed in Section 4.3.1. Notably, GPT-4o-mini exhibits stable performance in the GPA dimension across diverse graph group types, highlighting its strong generalization capability. In contrast, QVQ-72B-Preview

shows significantly lower scores on structurally complex and abstract graphs, underscoring its limitations in interpreting and reasoning over intricate graph structures. Interestingly, we observe that certain models exhibit stronger performance on the heterogeneous-graph type compared to some individual single-type graph types. This suggests that the presence of diverse structural forms may provide complementary cues that facilitate more effective cross-graph reasoning and the extraction of task-relevant information.

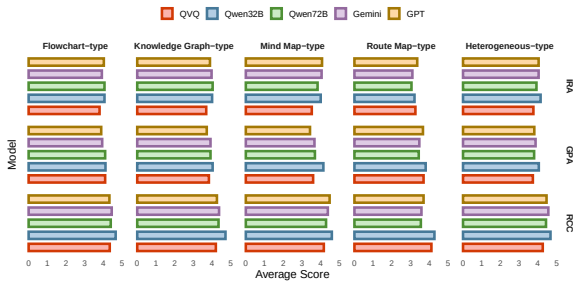


Figure 16: Average scores of different models across three evaluation dimensions—GPA, RCC and IRA each graph group type.

To further investigate fluctuations in model behavior, we conducted a variance analysis of performance across graph group types. The box plots of all models across the three evaluation dimensions for each graph group type are shown in Figure 17.

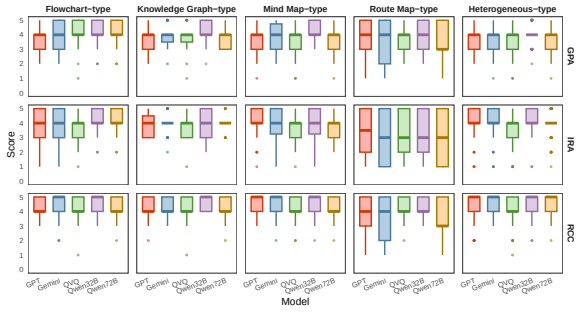


Figure 17: Box plots showing the distribution of model scores across five graph group types (columns) and three evaluation dimensions: GPA, IRA, and RCC (rows). Each boxplot summarizes the model's score distribution on a 5-point scale. For each model under a given condition, the top and bottom edges of the box represent the upper (75th percentile) and lower (25th percentile) quartiles of the score distribution, respectively. The horizontal line inside the box indicates the median score. The vertical lines extending from the box show the full range of non-outlier values. Dots outside this range represent outlier scores that deviate significantly from the main distribution.

We observed that all models exhibit notably higher variance in IRA when processing route maps. We attribute this to the unique characteristics of route maps compared to more structured formats like flowcharts or knowledge graphs. Specifically, route maps often involve spatial localization and path choices. Their nodes typically represent geographic locations or landmarks, while instructions tend to rely on spatially contextual, such as "go from A to B" or "turn left at the main road" (as illustrated in the example shown in Figure 18). Current

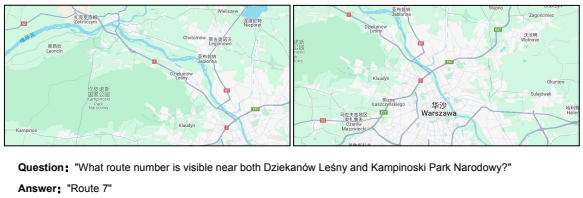


Figure 18: An example of a spatially grounded question in the route map setting. The question requires identifying a route number that appears near both "Dziekanów Leśny" and "Kampinoski Park Narodowy", demanding spatial localization and visual proximity reasoning. To answer correctly, the model must scan across multiple map regions, locate both landmarks, and detect the overlapping route label ("Route 7").

VLMs, however, are primarily optimized for logical reasoning and entity-based structures, lacking mechanisms for fine-grained spatial planning and directional awareness. This limitation introduces substantial randomness in how models interpret and execute route-based instructions, leading to high sample-level performance variance.

In contrast, we found that mind maps—due to their deep structural branching and high semantic density—emerge as one of the most discriminative graph types, revealing pronounced performance gaps between models. This underscores their diagnostic value for multi-graph reasoning benchmarks.

To further assess whether certain graph types induce "luck-driven" performance or contain outlier cases, we conducted a skewness analysis, which measures the asymmetry of a distribution; a value close to zero indicates a symmetric distribution, while positive or negative values indicate right- or left-skewed distributions, respectively. The sample skewness is computed as:

$$\text{Skewness} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^3 \qquad (10)$$

where $\bar{x}$ is the sample mean, s is the standard de-

viation, and n is the number of observations. The corresponding heatmap visualization is shown in Figure 19.

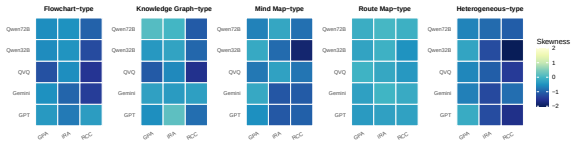


Figure 19: Skewness heatmap of model performance across five graph group types (columns) and three evaluation dimensions: GPA , IRA, and RCC. Each cell shows the skewness of the score distribution for a given model (rows) on a specific dimension. Warmer colors indicate higher positive skewness, while darker colors represent negative skewness.

Despite the high variance observed for route maps, the skewness of IRA scores across all models is consistently close to zero. This suggests that the observed variance is not driven by a few extreme samples but rather reflects a systemic bottleneck in spatial reasoning capabilities.

Therefore, while route maps pose a high-variance challenge, their stable distributional characteristics also make them suitable for benchmarking robustness. They provide both discriminative power and reliability, making them a valuable component in the design of future multi-graph evaluation tasks.

D.3.2 Evaluation Across Task Difficulty Levels

To further examine the generalization capabilities of each model under varying levels of task complexity, we conducted a systematic statistical analysis of their performance across three difficulty levels (easy, medium, difficult) and three evaluation dimensions, as shown in Figure 20.

Figure 20: Average scores of different models across three evaluation dimensions—GPA, RCC and IRA—under each difficulty level.

We observe that certain models, such as Qwen2.5-VL-32B-Instruct and GPT-4o-mini, exhibit a performance decline as task difficulty increases. This trend suggests that some models still struggle with multi-graph joint reasoning tasks involving complex structural information, extended reasoning paths, or semantically ambiguous inputs. In contrast, other models show relatively stable performance across different difficulty levels, and even achieve higher scores on more challenging tasks. This may be attributed to their stronger generalization ability, which allows them to better leverage the clearer visual-textual alignments and structural cues present in complex tasks, thereby enhancing reasoning performance under higher difficulty.

To complement the main difficulty-based evaluation, we further analyzed the robustness of each model when facing increasing task complexity in multi-graph joint reasoning.

We introduce a simple yet effective robustness metric: the difference between the average score on easy questions and that on difficult questions. Formally, let S_{easy} and $S_{\text{difficult}}$ denote a model's average score on easy and difficult questions respectively. The robustness score is defined as:

$$\text{Robustness} = S_{\text{difficult}} - S_{\text{easy}} \qquad (11)$$

A value closer to zero indicates that the model maintains relatively consistent performance under increased task difficulty, suggesting stronger resilience to complex structural and semantic conditions. Figure 21 summarizes the robustness scores

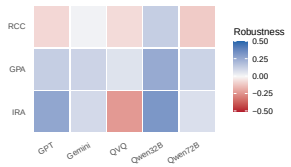


Figure 21: Robustness scores of all evaluated models across three reasoning dimensions. Each value represents the difference in average scores between easy and difficult questions. Lower absolute values indicate stronger robustness to increasing task difficulty.

of all evaluated models.

To further visualize model behavior across difficulty levels and graph types, we provide a cross-dimensional heatmap in Figure 22. This view enables a more fine-grained understanding of how each model handles variation in structural complexity and task formulation.

Figure 22: Cross-dimensional heatmaps illustrating model performance across varying difficulty levels and graph types under three evaluation dimensions. Each tile indicates the average score achieved by a model for a given evaluation dimension, graph type, and difficulty level. This visualization highlights model robustness and sensitivity to structural complexity across different reasoning competencies.

D.4 Variance Analysis of Model Scores Before and After Fine-Tuning

It is worth noting that although DeepSeek-VL-1.3B-Chat demonstrates significant performance improvement after fine-tuning, its score variance also increases, indicating a certain degree of inconsistency in its performance across samples. This fluctuation may stem from the model's differentiated adaptability to various task structures during fine-tuning. Specifically, for tasks with clear structure and well-defined goals—such as map-based localization—the model is able to learn stable reasoning patterns more effectively, resulting in notable performance gains. In contrast, for multi-graph tasks characterized by high information density and complex inter-graph logical relationships, the model may exhibit deviations in reasoning path selection or key information extraction, which in turn leads to declines in comprehension and answer quality. In the following section, we provide examples to illustrate this phenomenon in detail.

As illustrated in Figure 23, Sample 1 presents a task that requires identifying the location directly north of "Mbemba" based on two interrelated maps. This sample is designed to evaluate the model's ability in multi-graph spatial orientation reasoning. Before fine-tuning, the model exhibited severe recognition errors and hallucinations: it generated a fabricated place name, "Kigoma," and even introduced irrelevant reasoning about "river flow direction," which was not present in the images. These issues indicate major deficiencies in graph structure understanding and reasoning path construction, resulting in the lowest possible score (1 out of 5) across all three evaluation dimensions. After fine-tuning, however, the model accurately identified and returned the correct place name, "Ganda-Sundi." While the response was brief, it fully satisfied the requirements of all evaluation dimensions and matched the reference answer provided by GPT-4o. Consequently, the model achieved full marks (5 out of 5) in all dimensions. This case demonstrates that fine-tuning significantly improved the model's graph understanding and structured reasoning capabilities, particularly enhancing its generalization and robustness in spatial reasoning tasks involving geographic orientation.

In contrast, as shown in Figure 24, Sample 2 involves a question that asks the model to identify the immediate subsequent nodes following the initial nodes in two separate flowcharts. Before fine-tuning, although the model exhibited some deficiencies in graph understanding—failing to recognize the "providing the menu" node in the second flowchart and instead misinterpreting it as a synonymous expression of the "receiving parts" node in the first flowchart—it was still able to correctly infer the next step in the first graph. After fine-tuning, however, the model produced clearly misaligned or irrelevant flowchart nodes as its answer. This suggests that the fine-tuning process may have inadvertently impaired the model's original ability to parse graph structures, or introduced a tendency to follow incorrect reasoning paths.

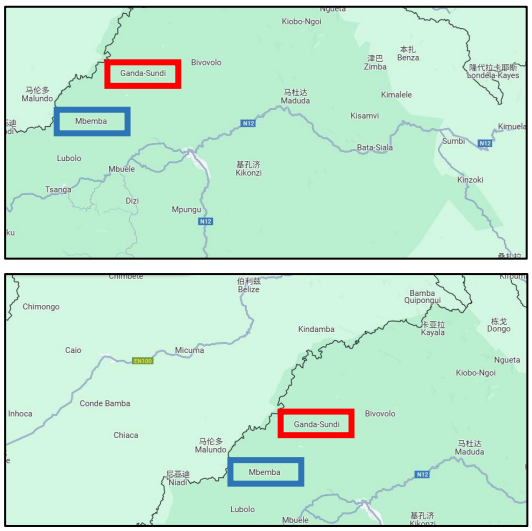
Therefore, we speculate that fine-tuning enhances the model's responsiveness to specific types of multi-graph reasoning tasks, but at the same time amplifies its performance disparities across different task structures, leading to an increase in score variance.

E Compute Resources

To support reproducibility and scalability analysis, we provide details of the compute environment for all experiments conducted in this work.

Compute Environment:

- **GPU:** NVIDIA A800 80GB × 1
- **CPU:** 14 vCPU Intel(R) Xeon(R) Gold 6348 @ 2.60GHz
- **RAM:** 100 GB
- **CUDA:** 12.4
- **OS/Container:** Ubuntu 22.04



Question:
Which place is directly north of Mbemba in both maps?
Answer:
Ganda-Sundi.
Model Response (Before Finetune):
The place directly north of Mbemba in both maps is Kigoma. This can be determined by looking at the map
where Mbemba is located, and then following the river that flows northwards from Mbemba to Kigoma. The river
is labeled as "Kigoma" on the map, and it is the only river
that flows north from Mbemba. The other rivers labeled
on the map are flowing southwards from Mbemba, as indicated by the labels "Mwembeshi" and "Mwembezi" on the map.
Model Response (After Finetune):
Ganda Sundi.

Figure 23: An example (Sample 1) illustrating improved spatial reasoning after fine-tuning. The model corrects its hallucinated response and successfully identifies the correct location "Ganda-Sundi" based on the directional relationship across two related maps.

- **Python Version:** 3.12
- **Frameworks:** PyTorch 2.5.1, ms-swift

All experiments were conducted using a single A800 GPU. No distributed or multi-node training was involved.

E.1 Broader Societal Impacts

This work introduces a benchmark for multi-graph reasoning with vision-language models (VLMs), constructed entirely from publicly available or synthetic content to support reproducible academic research.

The benchmark promotes advances in structured visual reasoning and enables fair, systematic evaluation across models, with potential applications in education, science, and healthcare.

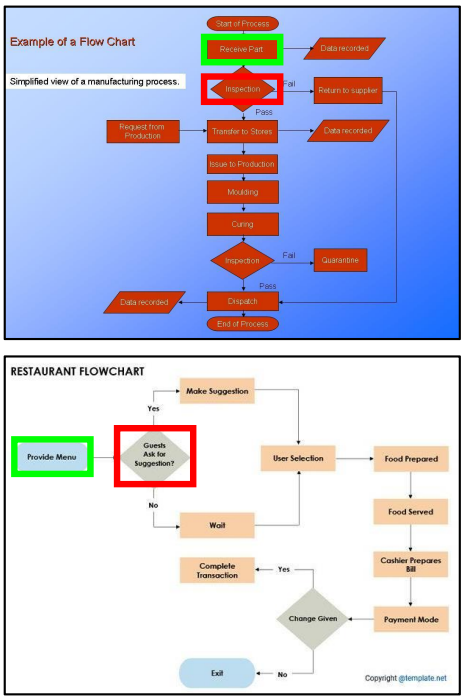
While released under the permissive MIT License, we encourage responsible use. Misapplication in high-stakes settings without proper validation may lead to misleading conclusions or unintended outcomes.

Users are advised to carefully assess real-world applicability before deployment.

F Dataset Safeguards

To ensure the safety and ethical integrity of our released dataset, we conducted a manual screening process. While most of the data is synthetically generated or manually created, a small portion was collected from public web sources.

All web-crawled images were individually reviewed by human annotators to ensure they do not contain personal, sensitive, or inappropriate content. This manual screening process ensures the dataset is safe and suitable for academic use under the MIT License.



Question:
In both the manufacturing process and the restaurant flowcharts, what follows immediately after the initial step of receiving parts or providing the menu?

Answer:
In the manufacturing process flowchart, after receiving parts, the next step is Inspection. In the restaurant flowchart, after providing the menu, the next step is determining if guests ask for suggestions.

Model Response (Before Finetune):
The immediate step after receiving parts or providing the menu in the manufacturing process is to inspect the parts. This is depicted in the flowchart where the "Inspection" step is directly connected to the "Receiving Parts" step. The visual clue supporting this statement is the arrow connecting the "Receiving Parts" step to the "Inspection" step in the flowchart, indicating the direct connection between these two actions.

Model Response (After Finetune):
In the manufacturing process flowchart, after receiving parts, the next step is to transfer to the store, followed by a move to the receiving department. In the restaurant flowchart, after receiving the food request, the next step is to enter the kitchen.

Figure 24: Sample 2 illustrates the performance shift of the model on a flowchart structure understanding task before and after fine-tuning. While the pre-finetuned model exhibits partial comprehension errors, it still manages to infer the correct subsequent node in one of the diagrams. In contrast, the post-finetuned model outputs misaligned or irrelevant nodes, suggesting a degradation in structural parsing capability after fine-tuning.