Benchmarking and Improving Large Vision-Language Models for Fundamental Visual Graph Understanding and Reasoning

Yingjie Zhu[1,2], **Xuefeng Bai**[1*], **Kehai Chen**[1,2], **Yang Xiang**[2*], **Jun Yu**[3], **Min Zhang**[1,2]
[1]Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen, China
{baixuefeng,chenkehai,yujun,zhangmin2021}@hit.edu.cn
zhuyj@stu.hit.edu.cn, xiangy@pcl.ac.cn

Abstract

Large Vision-Language Models (LVLMs) have demonstrated remarkable performance across diverse tasks. Despite great success, recent studies show that LVLMs encounter substantial limitations when engaging with visual graphs. To study the reason behind these limitations, we propose VGCURE, a comprehensive benchmark covering 22 tasks for examining the fundamental graph understanding and reasoning capacities of LVLMs. Extensive evaluations conducted on 14 LVLMs reveal that LVLMs are weak in basic graph understanding and reasoning tasks, particularly those concerning relational or structurally complex information. Based on this observation, we propose a structure-aware fine-tuning framework to enhance LVLMs with structure learning abilities through three self-supervised learning tasks. Experiments validate the effectiveness of our method in improving LVLMs' performance on fundamental and downstream graph learning tasks, as well as enhancing their robustness against complex visual graphs.[1]

1 Introduction

Graphs serve as a fundamental data structure across a wide range of domains, including social network analysis (Schweimer et al., 2022), recommendation systems (Zhang et al., 2023), knowledge graphs (Zhang et al., 2024b), chemistry (Cao et al., 2024), biomedical molecules (Liu et al., 2023) and semantic reasoning (Bai et al., 2022). Existing methods have achieved great success in enhancing understanding and reasoning abilities in graph-based tasks (Kim et al., 2023a; Chen et al., 2024). However, these approaches typically focus on specific graph types or tasks, posing challenges in designing versatile systems that are suitable for various tasks and graphs across diverse domains.

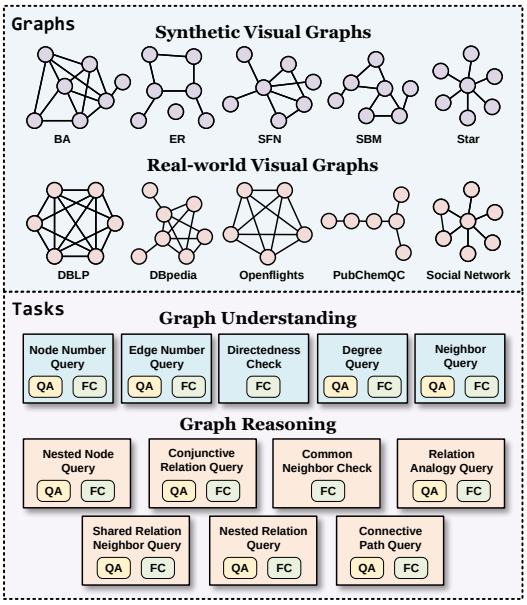


Figure 1: Overall of our VGCURE benchmark.

Recently, Large Vision-Language Models (LVLMs) have exhibited outstanding performance across a wide range of downstream tasks by unifying various inputs in the form of images and processing them with human-like understanding and reasoning abilities (Zhu et al., 2024; Zheng et al., 2024; Zhang et al., 2025). This triggers a growing interest in employing LVLMs for graph learning problems, as the vision modality offers a natural and intuitive way for comprehending structural information and facilitating general graph reasoning (Poklukar et al., 2022). However, recent studies (Wei et al., 2024; Li et al., 2024c; Ai et al., 2024) reveal significant challenges for LVLMs in graph-based learning tasks, where LVLMs achieve less than 15% accuracy on mathematical graph reasoning tasks, markedly below their performance in image and text reasoning (Li et al., 2024c). Therefore, a natural challenge arises: *why do LVLMs fail in graph learning, and how to enhance LVLMs to process graphs like professionals?*

To address this challenge, we begin by identi-

*Corresponding Author
[1]Our dataset and code are available at https://github.com/AAAndy-Zhu/VGCure.

Benchmark	Evaluation Topic	# Tasks	Graph Type	Anonymity	# Scale
VisionGraph	Graph Theory Problems	10	Synthetic	✓	3,000
Ai et al. (2024)	Multi-hop Reasoning	N/A	Real-world	✗	1,355
GVLQA	Graph Theory Problems	7	Synthetic	✓	157,896
our **VGCURE**	Fundamental Graph Understanding and Reasoning	22	Synthetic & Real-world	✓	223,646

Table 1: Comparisons among different visual graph analysis benchmarks for LVLMs, where *# Tasks* and *# Scale* represent the number of task types and test samples, respectively.

fying a research gap in existing evaluations that limit our understanding of LVLMs. Current work primarily focuses on graph theory problems or multi-hop reasoning tasks (Wei et al., 2024; Ai et al., 2024), which are complex and require a combination of diverse cognitive abilities. Thus it remains uncertain whether LVLMs have strong **fundamental understanding and reasoning capabilities** in processing visual graphs—such as recognizing the basic components of a graph and making basic logical inferences based on graph data—which is crucial for determining whether the limitations stem from a lack of fundamental abilities or other higher-order capabilities.

To this end, we present the **V**ision **G**raph **C**omprehensive **U**nderstanding and **RE**asoning benchmark, **VGCURE**, designed to thoroughly evaluate the fundamental understanding and reasoning capabilities of LVLMs on visual graphs. As shown in Fig.1, VGCURE evaluates the fundamental capabilities of LVLMs diverse challenges, including 9 graph understanding tasks and 13 graph reasoning tasks. Moreover, VGCURE features 10 anonymized graph structures from both synthetic and real-world sources, offering a robust testbed for assessing LVLMs' proficiency in handling diverse graphs. Experiments on 14 representative LVLMs show that current LVLMs exhibit weak fundamental understanding and reasoning capabilities on visual graphs, especially in capturing relational information and dealing with structurally complex visual graphs.

Motivated by the above observations, we further introduce MCDGRAPH, a novel structure-aware fine-tuning framework to enhance structure learning capabilities of LVLMs through three self-supervised learning tasks: 1) masked graph infilling, 2) contrastive graph discrimination, and 3) graph description. Experiments show that MCDGRAPH significantly improves LVLMs' graph understanding and reasoning abilities, especially on edge-related tasks and graph-related downstream tasks. Further analysis demonstrates that our method also enhances LVLMs' robustness and generalization to visual graphs with complex structure and unseen styles. The contributions of this work can be summarized as follows:

- We introduce VGCURE, a comprehensive benchmark to systematically evaluate LVLMs' fundamental understanding and reasoning abilities on visual graphs.
- Through extensive experiments on 14 LVLMs, we reveal LVLMs' limitations in basic graph understanding and reasoning, especially for tasks concerning relational or structural information.
- We propose a self-supervised framework to enhance LVLMs' ability to capture structural information in visual graphs. Experiments validate its effectiveness on both fundamental and downstream graph learning tasks.

2 The VGCURE Benchmark

To evaluate LVLMs' fundamental graph understanding and reasoning capabilities, we introduce VGCURE, a large-scale multimodal graph benchmark with 22 challenging tasks. VGCURE features 10 graph types, both synthetic and real-world, to assess LVLMs' performance on diverse graphs. The graphs are anonymized to minimize the impact of pre-existing LLM knowledge on core reasoning abilities, promoting *knowledge-free reasoning* (Hu et al., 2024). Tab.1 compares VGCURE with three recent benchmarks. It is evident that VGCURE excels in fundamental graph understanding and reasoning capabilities. Furthermore, VGCURE offers a comprehensive evaluation through more varied graph types, tasks, and test samples.

2.1 Graph Structure Generation

We begin by collecting a wide variety of graph structures for generating visual graphs and challenging tasks. Following Fatemi et al. (2024), we first employ *NetworkX* (Hagberg et al., 2008) to generate a diverse set of random synthetic structures, including Erdős-Rényi (ER) graphs (Erdős and Rényi, 1959), scale-free networks (SFN) (Barabási and Albert, 1999), Barabási–Albert (BA) model (Albert and Barabási, 2002), stochastic

Task	QA sample	FC sample (Label)
NNu	Q: How many nodes are there in this graph? A: 12	There are 12 nodes in this graph. (True) There are 17 nodes in this graph. (False)
EN	Q: How many edges are there in this graph? A: 15	There are 15 edges in this graph. (True) There are 17 edges in this graph. (True)
DC	-	This graph is a directed graph. (True) This graph is an undirected graph. (True)
DQ	Q: What is the degree of E9 in this graph? A: 1	The degree of E9 in this graph is 1. (True) The degree of E9 in this graph is 2. (False)
NQ	Q: Which nodes are out-neighbors of E6 in this graph? A: [E3]	E3 is a out-neighbors of E6 in this graph. (True) E7 is a out-neighbors of E6 in this graph. (False)
NN	Q: Which entities are R6 of the entity that is R5 of E3? A: [E4, E7]	E4 is R6 of the entity that is R5 of E3. (True) E1 is R6 of the entity that is R5 of E3. (False)
CR	Q: Which entities are R4 of E9 as well as R8 of E1? A: [E2]	E2 is R4 of E9 as well as R8 of E1. (True) E1 is R4 of E9 as well as R8 of E1. (False)
CN	-	E8 and E1 share a common out-neighbor, i.e., common head entity. (True) E10 and E12 share a common out-neighbor, i.e., common head entity. (False)
RA	Q: Which entities are connected to E3 via the same relation from E3 to E1? A: [E2]	E2 is connected to E3 via the same relation from E3 to E1. (True) E6 is connected to E3 via the same relation from E3 to E1. (False)
SRN	Q: Which entities are both R2 of E10? A: [E4, E12]	E4 and E12 are both R2 of E10. (True) E4 and E3 are both R2 of E10. (False)
NR	Q: What is the relation from the entity that is R5 of E3 to E2? A: [R8]	E2 is R8 of the entity that is R5 of E3. (True) E2 is R4 of the entity that is R5 of E3. (False)
CP	Q: Is there a path from E3 to E4? A: Yes. The paths are [[E3, E1, E2, E4], [E3, E1, E4]]	There are 2 paths from E3 to E4. (True) There are 3 paths from E3 to E4. (False)

Table 2: Examples for each task in VGCURE. These samples all correspond to the graph shown in Fig.7(d).

block model (SBM) (Holland et al., 1983) and star graphs. In addition, we extract anonymized structures from real-world graphs in GraphArena (Tang et al., 2024), including DBLP, Social Network, DBpedia, Openflights and PubChemQC. All the entity and relation names in each graph are replaced with generic names to eliminate the impact of the model's internal knowledge on reasoning. After initializing the graph structure, we use the *Graphviz* (Ellson et al., 2002) to generate concise *directed* and *undirected* visual graphs.

2.2 Tasks Design

To thoroughly assess the abilities of LVLMs in fundamental graph structure understanding and reasoning, the proposed VGCURE encompasses the following categories of tasks. Tab.2 presents examples for each task.

Graph Understanding: The graph understanding tasks involve analyzing and extracting structural, relational, and property-based information from the visual graph, which aims to gain insights into the composition and topology of the graph, including its nodes, edges, connectivity, and the relationships among its components.

- **Node Number Query (NNu)**: Calculate the total number of nodes in the graph.
- **Edge Number Query (EN)**: Determine the total number of edges in the graph.
- **Directedness Check (DC)**: Verify whether the graph is undirected or directed.
- **Degree Query (DQ)**: Calculate the degree of the specified node.
- **Neighbor Query (NQ)**: Identify the nodes that are directly connected to the given node.

Graph Reasoning: The reasoning tasks focus on exploring the *knowledge-free reasoning* ability of LVLMs on visual graphs. To differentiate from graph understanding tasks, we designed a series of 2-hop reasoning tasks.

- **Nested Node Query (NN)**: Identify the entities linked to the given entity through a composite chain involving the given relations.
- **Conjunctive Relation Query (CR)**: Retrieve the entities satisfying both two independent relationship constraints with two distinct entities.
- **Common Neighbor Check (CN)**: Determine whether two entities share at least one common neighbor in a 2-hop relational path.
- **Relation Analogy Query (RA)**: Find the entities linked to a target entity via a relation identical to that links a given entity pair.
- **Shared Relation Neighbor Query (SRN)**: Identify the set of entities that are connected to the given entity through the given relation.
- **Nested Relation Query (NR)**: Identify the relation between a target entity and an intermediate entity obtained by traversing a specific relation path from a given entity.

Models	Understanding					Reasoning											
	NNu	EN	DQ	NQ		NN		CR		RA		SRN		NR		CP	
	Acc	Acc	Acc	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1	EM_F1	Label_Acc
SPHINX	21.03	11.38	15.45	12.84	29.45	6.05	9.81	7.28	16.44	14.62	21.69	11.62	26.93	1.54	4.99	0.68	**95.59**
Monkey	40.09	9.22	17.81	9.90	21.88	8.90	12.96	2.85	4.62	3.74	4.13	9.79	21.61	3.06	7.98	0.46	5.11
MiniGPT-v2	11.80	10.61	18.03	16.40	27.08	8.92	18.02	2.01	2.65	7.50	14.87	9.85	27.47	5.86	13.87	5.26	**95.59**
mPLUG-Owl3	28.38	8.84	6.86	20.32	51.16	8.68	11.39	5.42	11.81	11.07	14.81	6.48	18.06	2.03	0.33	10.87	74.97
LLaVA1.5-7B	14.53	7.56	30.14	11.43	21.25	1.80	0.21	2.22	0.13	6.36	9.37	5.38	7.73	0.84	2.23	2.63	78.64
LLaVA-NeXT	47.89	9.59	19.52	23.27	44.11	14.77	21.74	6.60	12.44	10.14	15.29	12.68	27.88	9.88	6.46	8.27	95.56
LLaVA-OV	23.47	3.08	23.80	15.00	39.12	10.74	19.87	7.50	16.61	9.85	19.30	10.21	30.99	1.04	1.36	9.69	94.83
LLaVA1.5-13B	17.08	7.62	26.33	15.68	32.66	6.08	7.83	4.31	13.23	6.52	10.80	7.48	12.86	4.21	6.21	5.91	82.41
InternLM-XC2.5	60.20	10.53	41.12	18.90	55.33	14.64	26.14	19.19	45.60	11.50	19.06	14.12	46.53	3.09	5.49	28.82	95.53
Llama3.2	77.31	9.39	35.56	18.79	42.34	**18.93**	29.43	21.90	55.00	**15.80**	27.51	20.79	61.30	**14.11**	**24.05**	20.80	94.70
Qwen-VL	42.45	9.66	20.56	10.95	21.25	11.44	15.51	7.93	18.30	12.48	17.36	11.11	22.79	5.43	4.01	0.00	4.48
Qwen2-VL	**97.80**	**16.38**	48.09	16.18	38.12	16.52	28.34	21.02	48.57	14.19	**27.96**	19.48	56.42	12.73	25.07	12.90	38.06
InternVL2	77.45	9.78	50.75	25.01	68.58	18.30	**30.82**	**24.87**	**59.31**	10.83	17.12	20.72	**59.99**	10.58	18.97	14.53	43.97
GPT-4o-mini*	89.20	15.40	**56.40**	**30.81**	**77.40**	17.01	29.98	22.33	53.15	15.47	24.81	**22.25**	59.48	8.62	13.48	**42.55**	92.40

Table 3: Model performance on QA samples across various tasks, where *EM_F1* is the macro F1 score calculated based on the exact match between the predicted path and the ground truth path, *Label_Acc* measures the accuracy of the model's prediction on whether a path exists or not. The best results are **bolded**.

- **Connective Path Query (CP)**: Determine the existence of *directed* paths or *shortest undirected* paths between two given entities, and retrieve all possible paths if they exist.

For each task, we construct **one QA sample** and **two fact checking (FC) samples** (with labels of True and False, respectively) automatically based on the template, except for CN and DC which have only fact checking samples due to the strong similarity between the two samples. The total number of final samples is 223,646. More details about the **design logic** and **statistics** of VGCURE can be found in Appendix A and Tab.9.

3 Benchmarking LVLMs on VGCURE

3.1 Experimental Setup

We conduct evaluation on 13 open-source LVLMs, including InternLM-XComposer2.5-7B (Zhang et al., 2024a), InternVL2-8B, Llama3.2-11B-Vision-Instruct, LLaVA1.5-7B (Liu et al., 2024), LLaVA1.5-13B (Liu et al., 2024), LLaVA-NeXT-7B (Li et al., 2024b), LLaVA-OneVision-7B (Li et al., 2024a), MiniGPT-v2 (Chen et al., 2023), Monkey (Li et al., 2024d), mPLUG-Owl3-7B (Ye et al., 2024), Qwen-VL (Bai et al., 2023), Qwen2-VL-7B-Instruct (Wang et al., 2024) and SPHINX (Lin et al., 2023). Meanwhile, we also evaluate the performance of the GPT-4o-mini, which is a strong closed-source LVLM[2]. Due to the high cost of GPT-4o-mini, we randomly select 50 graphs from each graph structure for testing. For all methods, the **zero-shot** setting is adopted during evaluation. More details can be found in Appendix B.

[2]We excluded GPT-4o as a baseline due to its high cost.

3.2 Main Result

Graph Understanding Tab.3 and Tab.4 present the evaluation results for question answering (QA) and fact checking (FC), respectively. We report the averaged results across various graph structures. In general, most LVLMs struggle to precisely understand the structural and relational information in visual graphs. In specific, (I) Among the graph understanding tasks, *Node Number Query (NNu)* and *Directedness Check (DC)* are the easiest for most LVLMs. This indicates that most LVLMs can accurately capture the number of nodes and directedness information within the visual graph. (II) All LVLMs struggle with *Edge Number Query (EN)* and *Neighbor Query (NQ)* tasks, with the highest accuracy of 16.38% and F1 score of 30.81% , respectively. This indicates that current LVLMs are weak in understanding relational and structural information. (III) Even with a similar number of parameters, the graph understanding abilities of open-source LVLMs vary significantly, with Qwen2-VL and InternVL2 showing better performance in both QA and FC samples. (IV) For the same task, LVLMs perform differently on QA and FC samples, likely due to different ways in reasoning and understanding required by each task (Thorne et al., 2018). (V) The closed-source LVLM, GPT-4o-mini, offers no significant advantages and even underperforms open-source LVLMs, especially on FC tasks.

Graph Reasoning Based on graph reasoning results in Tabs.3 and 4, it can be observed that, (I) Compared to graph understanding, the graph reasoning tasks are more challenging and the overall performance of LVLMs is worse on both

Models	Understanding										Reasoning													
	NNu		EN		DC		DQ		NQ		NN		CR		CN		RA		SRN		NR		CP	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
SPHINX	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.84	51.16	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00
Monkey	40.31	53.23	51.38	55.58	64.25	65.24	50.01	55.62	44.04	53.29	50.54	50.76	37.37	51.08	48.85	49.71	38.32	50.13	33.52	49.78	40.29	48.05	51.44	**58.87**
MiniGPT-v2	34.37	50.38	34.28	50.39	33.92	50.18	33.69	49.91	44.29	50.99	49.68	50.03	47.94	53.19	45.66	51.80	36.71	49.48	43.08	43.22	36.68	50.07	51.68	52.39
mPLUG-Owl3	37.53	50.80	31.16	38.59	73.04	74.76	39.84	46.92	33.33	50.00	34.75	49.86	46.11	52.04	33.96	51.20	44.80	48.54	47.35	53.10	36.65	49.30	37.34	50.27
LLaVA1.5-7B	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.84	51.16	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00
LLaVA-NeXT	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00	33.84	51.16	33.33	50.00	33.33	50.00	33.33	50.00	33.33	50.00
LLaVA-OV	39.13	52.69	33.79	49.68	65.02	68.49	36.87	51.49	37.12	51.70	46.83	53.37	39.66	52.41	34.09	51.23	39.64	48.89	44.12	52.19	48.19	52.19	33.35	49.05
LLaVA1.5-13B	60.31	63.26	50.78	57.09	87.74	87.92	33.33	50.00	36.39	49.31	47.65	50.31	35.93	50.94	34.68	49.31	33.33	50.00	33.47	50.03	35.18	43.68	44.55	52.91
InternLM-XC2.5	37.19	51.79	40.00	51.05	65.76	66.72	43.07	49.16	64.46	65.40	50.81	54.05	58.72	61.66	39.34	50.05	35.54	50.47	49.36	51.98	49.12	53.08	52.12	52.76
Llama3.2	87.66	**87.66**	47.26	49.67	43.27	51.83	42.23	51.39	65.87	66.03	53.47	56.40	62.79	63.74	39.00	50.30	**59.89**	**61.20**	48.82	49.38	**58.59**	**60.36**	45.65	49.32
Qwen-VL	32.30	47.72	32.27	47.56	9.50	10.49	31.04	44.54	36.69	49.97	27.07	32.57	15.55	17.99	31.68	45.30	27.83	29.13	32.71	33.70	34.72	41.94	36.37	42.63
Qwen2-VL	**76.50**	77.67	**68.26**	**68.27**	**94.92**	**94.94**	**64.32**	**67.40**	67.34	68.76	44.17	53.77	74.28	75.24	38.52	51.10	35.28	50.55	**57.07**	**59.18**	42.08	52.95	42.25	53.38
InternVL2	68.63	71.18	36.82	50.23	93.17	93.18	63.28	63.84	**72.81**	**73.27**	**62.46**	**63.62**	**75.37**	**76.18**	47.12	50.40	33.70	50.14	56.98	57.42	55.94	58.42	**54.71**	54.71
GPT-4o-mini*	64.49	67.00	39.85	52.30	90.52	90.60	42.03	52.90	48.77	56.80	50.77	51.51	62.17	62.82	**53.03**	**55.05**	36.50	50.78	54.87	54.89	44.37	51.11	51.39	54.50

Table 4: Model performance on FC samples across various tasks. The best results are **bolded**.

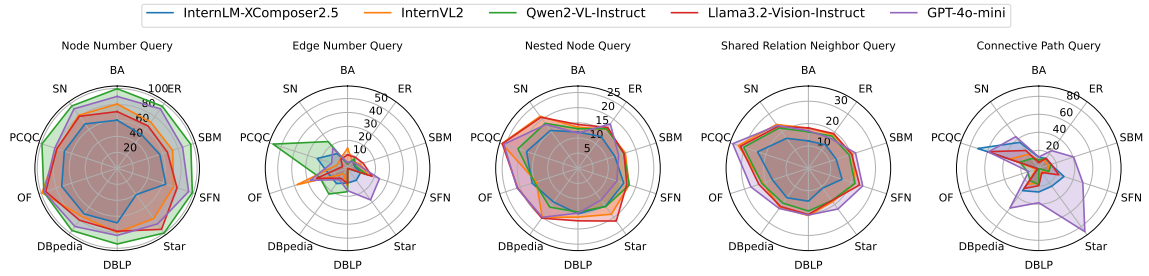


Figure 2: Model performance (F1/Acc) on representative QA samples across various graph structures and tasks , where OF, PCQC and SN denotes Openflights, PubChemQC and Social network, respectively.

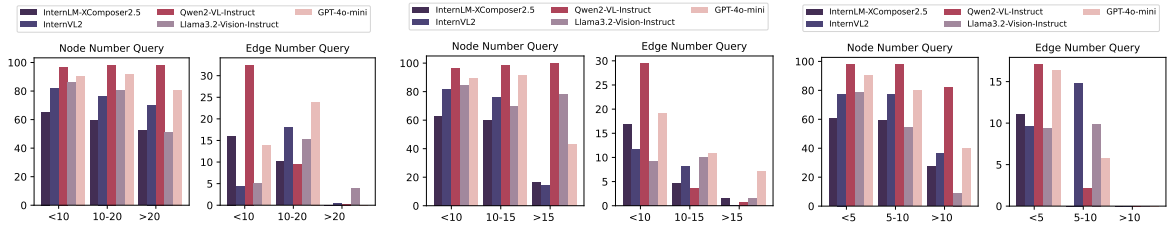
QA and FC samples. This may be a knock-on effect due to deficiencies in visual graph understanding. (II) Among all open-source LVLMs, InternVL2 and Llama3.2-Vision achieve better performance and LLaVA1.5-7B perform the worst on graph reasoning tasks. (III) All the LVLMs perform poorly on *Nested Relation Query (NR)* for both QA and FC samples, which is similar to the observation in the graph understanding task. This suggests that LVLMs are deficient in recognizing edges and understanding structural information within visual graphs. (IV) On the QA samples, the performance of different LVLMs on *Connective Path Query (CP)* varies widely. SPHINX, Monkey and Qwen-VL demonstrate almost no ability to recognize paths between two specific nodes in the visual graph. (V) Similarly, GPT-4o-mini underperforms in most tasks compared to open-source LVLMs, except for the *Connective Path Query (CP)* task.

3.3 Impact of Structures

Inspired by Fatemi et al. (2024), we explore how graph structure affects LVLMs' ability to understand and reason on visual graphs. Fig.2 compares the performance of the top five LVLMs on QA samples across various structures. Obviously, the graph structure significantly impacts LVLMs' performance on most tasks. Notably, all LVLMs perform well on *PCQC*, which has a simpler structure with fewer nodes and edges (averaging 5.45 nodes and 4.76 edges), and weaker on *BA*, which has the highest edge count in VGCURE (averaging 21.02 edges). Furthermore, on different tasks, the performance of LVLMs is affected differently by the graph structure. *EN* and *CP* show larger performance variations across graph structures, whereas *NNu* and *NN* show smaller differences. More results are shown in Figs.12 and 13, the overall trends are similar to above findings.

3.4 Impact of Complexity

We further discuss the impact of graph complexity on the LVLMs' ability to understand and reason over the visual graph by considering the number of edges, number of nodes, and average degree. Fig.3 shows a performance comparison of five LVLMs on QA samples across representative tasks and complexity levels. As complexity increases, LVLMs' performance declines, especially on *EN*, where results vary significantly. Besides, some LVLMs perform best at intermediate complexity, but struggle with more complex graphs. This reflects a balance between information richness and complexity of visual graphs, whereas higher complexity likely overwhelm the LVLMs' abilities to reason or generalize due to the complex information within large graphs. In addition, different complexity dimensions affect LVLMs'



(a) Number of Edges (b) Number of Nodes (c) Average Degree

Figure 3: Model performance (Acc) comparison on QA samples across various dimensions of complexity.

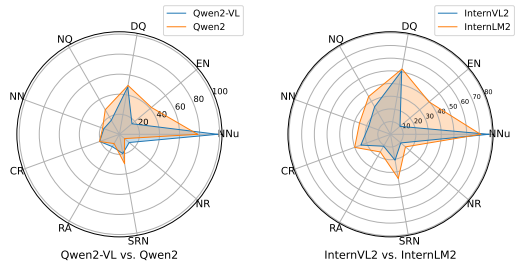


Figure 4: Results comparison between LVLM and LLM on QA samples across various tasks.

Error Type	R.M.	C.L.	S.H.	E.A.	Ot.G.	F.E.
Percentage	54%	7%	17%	9%	5%	8%

Table 5: Frequency of each error type.

performance on various tasks differently, such as *NNu*, where LVLMs are more influenced by the number of nodes and average degree than by edges. More results in Figs.14-19 show the similar trend.

3.5 Why LVLMs Fail on Fundamental Tasks?

To explore in depth whether the failure of LVLMs on the fundamental task is due to their weak **perception** or **other higher-order** ability, we compare the performance of LVLMs and the corresponding backbone LLMs that take the textual graph as input. As Fig.4 shows, despite the disparity in the performance of the individual models, LVLMs perform weaker than the corresponding backbone LLMs on both understanding and reasoning tasks. In particular, LVLMs' performance degrades the most on *Edge Number Query (EN)*, which also illustrates the lack of ability of LVLMs to capture structural information in the visual graph. This indicates that the failure of LVLMs is partly attributed to their **weaker visual graph perception**. In addition, the performance of the backbone LLMs on these tasks remains undesired, with all accuracies below 45%, this suggests that the backbone LLMs exhibit limited capabilities in understanding and reasoning on graph data. More details and results are presented in Appendix C.1.

3.6 Error Analysis

In addition, by analyzing a batch of results generated by Qwen2-VL and InternVL2 on QA samples in VGCURE, we find that their main errors can be grouped into the following six categories:

- **Relation Misunderstanding (R.M.)**: Failure to properly understand the relations between entities, leading to erroneous reasoning.
- **Complexity Limitation (C.L.)**: When faced with intricate or highly complex graph structures, LVLMs struggle to process and understand the information effectively, often resulting in incomplete or erroneous outputs.
- **Structural Hallucination (S.H.)**: Generating or perceiving structures that do not exist, leading to erroneous or misleading results that do not match the actual visual graph.
- **Entity-based Answering (E.A)**: Directly using the entities mentioned in the question as answers, thus ignoring deeper relation understanding or logical reasoning in the visual graph.
- **Off-target Generation (Ot.G.)**: Deviation from the task or misunderstanding of the question, leading to the generation of irrelevant answers.
- **Format Error (F.E.)**: The output of the model is incorrectly formatted or unexpected.

Examples of each error are shown in the Tab.11.

Error Distribution Analysis We also calculate the overall distribution of each error. The results are shown in Tab.5. It can be observed that, (I) Relation Misunderstanding appears most frequently due to LVLMs' limited capacity to effectively capture structural and relational information in the visual graph. (II) Structural Hallucination also appears frequently due to the inherent hallucination tendency of LLMs. (III) Despite demonstrating robust instruction-following capabilities, LVLMs remain prone to errors like Off-target Generation and Format Error. (IV) Complexity Limitation and Entity-based Answering also account for a

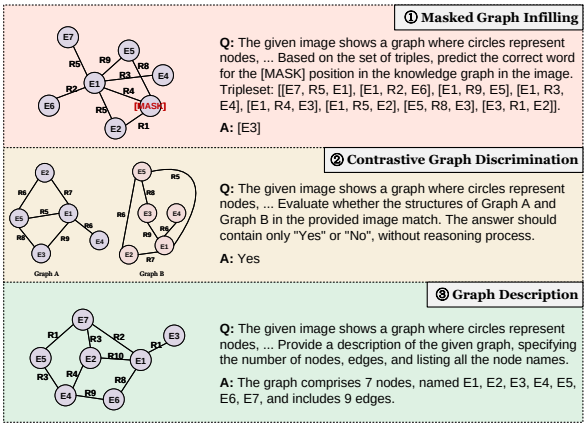


Figure 5: Overall illustration of MCDGRAPH.

certain percentage, i.e., LVLMs may be stumped when faced with overly complex visual graphs, or directly use the entities mentioned in the question as answers.

Primary Error Pattern Analysis We further analyze the primary error pattern of each task. We find that, (I) Generally, the primary error occurring in all tasks except the *Node Number Query* is Relation Misunderstanding. (II) Structural Hallucination occurs more frequently in *Nested Node Query*, *Conjunctive Relation Query, Relation Analogy Query*, *Shared Relation Neighbor Query*, and *Nested Relation Query* tasks. (III) Format Error is usually found in *Connective Path Query* task.

4 The MCDGRAPH Framework

To enhance the ability of LVLMs to understand and reason on visual graphs, we propose MCDGRAPH, a **self-supervised** fine-tuning framework designed to improve LVLMs' ability to capture structural and relational information within visual graphs. As illustrated in Fig.5, MCDGRAPH comprises three key tasks: **M**asked Graph Infilling, **C**ontrastive Graph Discrimination, and Graph **D**escription.

4.1 Task1: Masked Graph Infilling

For this task, we randomly mask either nodes or edges in a visual graph and challenge the model to predict the masked element based on the partially observed graph. Since anonymized visual graphs contain no semantic information, we also provide the corresponding text triples as input.

$$M = \text{LVLMs}(\hat{G}, I, T), \tag{1}$$

where $\hat{G}$, I, T denotes the masked graph, task instruction, and text triples of the original graph, respectively. This task encourages LVLMs to infer missing structure information, improving their ability to understand graph structure and the relationships between elements.

4.2 Task2: Contrastive Graph Discrimination

To further refine the LVLMs' understanding of graph structure, we introduce a contrastive learning task, which helps train the LVLMs to distinguish between two visual graphs that may either represent the *same graph with different layouts* or two *distinct graphs with similar layouts*.

$$Y = \text{LVLMs}(G_1, G_2, I), \tag{2}$$

where the answer $Y \in \{Yes, No\}$, G_1, G_2 denotes two graphs, and I is the task instruction. By learning how to perform structural reasoning and graphical isomorphism detection, this task aims to improve LVLMs by recognizing subtle structural differences between two visual graphs.

4.3 Task3: Graph Description

Graph Description task requires LVLMs to generate a textual description of a given visual graph, including the total number of nodes and edges, as well as the names of all the nodes in the graph,

$$D = \text{LVLMs}(G, I), \tag{3}$$

where D represents the description, and G, I denotes the input graph and task instruction, respectively. This task ensures that the LVLMs develop a clear understanding of the graph's composition, thereby enhancing their ability to interpret and summarize graph-based information.

5 Enhancing LVLMs with MCDGRAPH

5.1 Experimental Setup

We validate the effectiveness of MCDGRAPH on top two performing LVLMs on VGCURE, i.e., Qwen2-VL and InternVL2. We collect a new set of anonymized visual graphs **beyond VGCURE** with synthetic structures and automatically create 20k training samples for MCDGRAPH. To prevent catastrophic forgetting, we apply LoRA (Hu et al., 2022) to efficiently enhance the LVLMs' abilities while preserving their original performance. More details about training samples and implementation are available in Appendix D.

5.2 Results and Analysis

Main Results Tab.6 compares the performance of Qwen2-VL and InternVL2 before and after applying MCDGRAPH on both visual graph understanding and reasoning tasks. We can observe

Models	Understanding					Reasoning						
	NNu	EN	DC	DQ	NQ	NN	CR	CN	RA	SRN	NR	CP
	QA Samples											
Qwen2-VL	97.80	16.38	-	48.09	16.18	16.52	21.02	-	14.19	19.48	12.73	12.90
w MCDGRAPH	98.34 ↑	25.92 ↑	-	60.94 ↑	25.44 ↑	13.32	26.14 ↑	-	13.14	20.74 ↑	14.44 ↑	11.95
InternVL2	77.45	9.78	-	50.75	25.01	18.30	24.87	-	10.83	20.72	10.58	14.53
w MCDGRAPH	95.68 ↑	40.45 ↑	-	54.78 ↑	28.80 ↑	19.43 ↑	28.53 ↑	-	11.67 ↑	22.34 ↑	16.50 ↑	12.76
	FC Samples											
Qwen2-VL	76.50	68.26	94.92	64.32	67.34	44.17	74.28	38.52	35.28	57.07	42.08	42.25
w MCDGRAPH	89.58 ↑	65.80	95.84 ↑	77.10 ↑	79.75 ↑	60.71 ↑	83.07 ↑	53.90 ↑	53.48 ↑	64.17 ↑	64.12 ↑	60.11 ↑
InternVL2	68.63	36.82	93.17	63.28	72.81	62.46	75.37	47.12	33.70	56.98	55.94	54.71
w MCDGRAPH	76.55 ↑	71.98 ↑	90.04	56.83	80.98 ↑	73.14 ↑	80.23 ↑	52.09 ↑	52.81 ↑	59.07 ↑	69.03 ↑	45.82

Table 6: Model performance (Acc/F1/EM_F1 for QA and F1 for FC) on various tasks. ↑ indicates an improvement compared to the original model. The complete experimental results are shown in Tab.18 and 19.

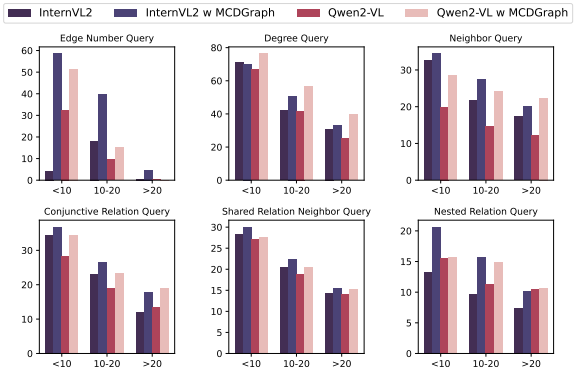


Figure 6: Model performance (F1/Acc) comparison on QA samples across representative tasks and edge ranges.

that (I) MCDGRAPH improves the performance of LVLMs on almost all tasks, demonstrating the effectiveness of the proposed method. (II) The improvement of LVLMs is particularly impressive on edge-related tasks, i.e., *Edge Number Query (EN)* and *Nested Relation Query (NR)*, that are relatively difficult for LVLMs. This suggests that MCDGRAPH enhances LVLMs' ability to capture structural information. (III) Although MCDGRAPH does not optimize for the tasks in VGCURE, it still shows obvious improvement in both QA and FC samples for most tasks in VGCURE. This demonstrates that the proposed method can improve the fundamental graph structure understanding capabilities of LVLMs, which leads to better performance on downstream tasks. We also present a case on *NR* to further understand the effectiveness of the proposed method, please refer to Appendix G for detailed information.

Ablation Study We also conduct ablation study on QA samples with Qwen2-VL and the results are shown in Tab.7. It can be observed that after removing different training tasks, although the LVLM may have a slight performance improvement on some tasks, it may lose certain

Models	Understanding				Reasoning					
	NNu	EN	DQ	NQ	NN	CR	RA	SRN	NR	CP
Qwen2-VL	97.80	16.38	48.09	16.18	16.52	21.02	14.19	19.48	12.73	12.90
w MCDGraph	98.34	25.92	60.94	25.44	13.32	26.14	13.14	20.74	14.44	11.95
- Masked	99.02	32.11	57.75	22.84	14.15	23.29	15.32	19.15	5.01	24.61
- Contrastive	99.64	28.70	64.02	25.35	15.16	25.13	11.83	21.11	15.20	5.33
- Description	2.00	13.27	57.80	23.09	14.14	23.47	10.16	19.87	16.99	8.41

Table 7: Ablation study on across various tasks.

capabilities causing its performance to plummet on some task (as evidenced by the red results in the Tab.7). Meanwhile, all three self-supervised tasks enhance the LVLMs' ability to capture structural and relational information from different dimensions, complementing each other in order to comprehensively improve the LVLM's overall performance on all tasks.

Performance on Varying Complexity Fig.6 illustrates the performance improvements of MCDGRAPH on LVLMs across varying graph complexities on six representative tasks in VGCURE. Fine-tuning with MCDGRAPH consistently improves performance across most tasks and complexity levels, demonstrating its its effectiveness in enhancing the LVLMs' ability to understand and reason over visual graphs. For simpler graphs, the improvement is smaller, where LVLMs already perform well, but as complexity increases, the performance gap between fine-tuned and baseline models becomes more pronounced, highlighting MCDGRAPH's importance in handling more complex visual graphs. Due to the space limit, the results for other dimensions are shown in Figs.20-25 in the Appendix, with conclusions similar to those above.

Generalization of MCDGRAPH To validate the generalization of our method, we regenerate 50 visual graphs with different **visual styles** and **naming conventions** of nodes and edges from those in VGCURE for each graph structure. The details and results are presented in Appendix E.

Model	VisionGraph			FACTKG			
	Connect	Cycle	MaxFlow	Accuracy	Precision	Recall	F1
Qwen2-VL	55.8	52.88	1.72	79.60	81.18	79.13	79.13
w MCDGRAPH	53.37	52.88	**5.17**	**80.15**	**81.71**	**79.69**	**79.71**
InternVL2	46.9	52.88	6.9	79.41	80.93	78.95	78.95
w MCDGRAPH	**54.72**	52.88	**8.62**	**79.78**	**81.34**	**79.32**	**79.33**

Table 8: Model performance on downstream tasks.

5.3 Results on Downstream Reasoning Tasks

To further demonstrate the scalability and applicability of our method, we evaluate MCDGRAPH on graph-related downstream reasoning tasks.

VisionGraph We first evaluate the performance of MCDGRAPH on three representative graph theory problems in VisionGraph (Li et al., 2024c). As the results shown in Tab.8, MCDGRAPH generally improve the performance of LVLMs on these tasks, especially for the relatively difficult *Maximum Flow task*. This confirms the effectiveness and scalability of our method.

FACTKG We then evaluate MCDGRAPH on FACTKG (Kim et al., 2023b), a knowledge graph-based fact verification dataset collected from **real-world data**. As Tab.8 shows, MCDGRAPH gives consistently better performance than LVLMs on FACTKG, suggesting that the proposed method can improve LVLMs in real-world graph-related tasks. Note that FACTKG not only requires fundamental visual graph understanding and reasoning abilities but also relies on LVLMs' understanding of semantic and logical relationships between entities, which our method does not address. Therefore, the improvement of LVLMs is not as significant as that of the graph theory problems. We also evaluate our method on two **general vision reasoning tasks** and the results are available in Appendix F.

6 Related Work

Multimodal Benchmark for Graphs

Li et al. (2024c) and Wei et al. (2024) introduce VisionGraph and GVLQA, respectively, for testing the problem-solving capabilities of LVLMs in graph theory. Both of them contain numerous synthetic visual graphs and complex graph theory problems. Besides, Ai et al. (2024) propose a novel instruction-following benchmark for multimodal graph understanding and reasoning, which contains a number of real-world graph images with diverse structures across various domains. However, these benchmarks focus on specific downstream tasks where LVLMs perform poorly. The goal of VGCURE is to assess LVLMs' fundamental understanding and reasoning abilities on visual graphs to identify the reasons for their failures.

Boosting LVLMs for Visual Graph Reasoning Li et al. (2024c) propose a Description-Program-Reasoning (DPR) chain to enhance logical accuracy through graphical structure description and multi-step reasoning. Wei et al. (2024) introduce GITA, an end-to-end framework integrating visual information into instruction-based graph reasoning. Additionally, Deng et al. (2024) present GraphVis, which uses curriculum fine-tuning for training LVLMs on feature recognition and visual graph QA tasks. Unlike current methods, our MCDGRAPH is a general-purpose, self-supervised approach that improves the fundamental understanding and reasoning of LVLMs on visual graphs, making it adaptable to most graph-related downstream tasks.

Graph Benchmarks for GNNs Rozemberczki et al. (2021) construct Wikipedia-based graphs with pages as nodes and hyperlinks as edges. Hu et al. (2020) present realistic datasets spanning social, biological, molecular, code, and knowledge graphs. Mernyei and Cangea (2020) propose Wiki-CS, a GNN benchmark based on Computer Science articles. Morris et al. (2020) release TUDataset, covering 120 datasets across multiple domains for graph classification and regression. To address homophily limitations, Lim et al. (2021) introduce large-scale, non-homophilous graph datasets. Dwivedi et al. (2022) propose the Long Range Graph Benchmark to evaluate models on long-range interaction reasoning tasks. Different from these graph benchmarking approach, the proposed VGCure focuses on exploring the fundamental graph understanding and reasoning capabilities of LVLMs, showcasing their potential to unify multimodal information processing through a unified visual learning paradigm.

7 Conclusion

This paper introduces VGCURE, a comprehensive benchmark comprising 22 tasks to evaluate LVLMs' fundamental understanding and reasoning capabilities on visual graphs. Experiments on 14 LVLMs reveal significant limitations, especially in capturing structural information. To this end, we propose MCDGRAPH, a structure-aware self-supervised method to enhance open-source LVLMs' structure learning abilities. Extensive experiments validate the effectiveness of our method across a wide range of graph-related tasks.

Limitations

- **Complexity of Visual Graphs.** Due to the limitations of current LVLMs' performance on visual graph tasks, we restrict the number of nodes in the synthetic graph structure to between 7 and 15, potentially limiting the exploration and improvement of the LVLMs' performance on more complex visual graphs.
- **Experiments on Larger LVLMs.** Due to limited resources, the majority of our experiments are performed only on LVLMs with around 7B parameters, lacking performance evaluation and improvement of larger models with more parameters.

Acknowledgments

We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions. This work was partly supported by the National Natural Science Foundation of China (62406091, 62276077, 62125201, U23B2055, U24A20328, U24B20174, 62350710797), the National Science and Technology Innovation 2030 Major program (2024ZD01NL00101), and Shenzhen Science and Technology Program (KQTD20240729102154066, ZDSYS20230626091203008).

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.

Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science*, 286(5439):509–512.

He Cao, Yanjun Shao, Zhiyuan Liu, Zijing Liu, Xiangru Tang, Yuan Yao, and Yu Li. 2024. PRESTO: Progressive pretraining enhances synthetic chemistry outcomes. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10197–10224.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Runjin Chen, Tong Zhao, AJAY KUMAR JAISWAL, Neil Shah, and Zhangyang Wang. 2024. LLaGA: Large language and graph assistant. In *Forty-first International Conference on Machine Learning*.

Yihe Deng, Chenchen Ye, Zijie Huang, Mingyu Derek Ma, Yiwen Kou, and Wei Wang. 2024. Graphvis: Boosting LLMs with visual knowledge graph integration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. 2022. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340.

John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull. 2002. Graphviz—open source graph drawing tools. In *Graph Drawing: 9th International Symposium, GD 2001 Vienna, Austria, September 23–26, 2001 Revised Papers 9*, pages 483–484. Springer.

Paul Erdős and Alfréd Rényi. 1959. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. *arXiv preprint arXiv:2406.09170*.

Aric Hagberg, Pieter J Swart, and Daniel A Schult. 2008. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Peng Hu, Sizhe Liu, Changjiang Gao, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. 2024. Large language models are cross-lingual knowledge-free reasoners. *arXiv preprint arXiv:2406.16655*.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023a. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9410–9421.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023b. FactKG: Fact verification via reasoning on knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16190–16206, Toronto, Canada. Association for Computational Linguistics.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024b. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

Yunxin Li, Baotian Hu, Haoyuan Shi, Wei Wang, Longyue Wang, and Min Zhang. 2024c. Visiongraph: Leveraging large multimodal models for graph theory problems in visual context. In *Forty-first International Conference on Machine Learning*.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024d. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26763–26773.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in neural information processing systems*, 34:20887–20902.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. 2023. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023. MolCA: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15623–15638.

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

Péter Mernyei and Cătălina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*.

Petra Poklukar, Miguel Vasco, Hang Yin, Francisco S. Melo, Ana Paiva, and Danica Kragic. 2022. Geometric multimodal contrastive representation learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 17782–17800.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014.

Christoph Schweimer, Christine Gfrerer, Florian Lugstein, David Pape, Jan A. Velimsky, Robert Elsässer, and Bernhard C. Geiger. 2022. Generating simple directed social network graphs for information spreading. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1475–1485.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162.

Jiasheng Si, Yibo Zhao, Yingjie Zhu, Haiyang Zhu, Wenpeng Lu, and Deyu Zhou. 2024a. CHECKWHY: Causal fact verification via argument structure. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15636–15659.

Jiasheng Si, Yingjie Zhu, Wenpeng Lu, and Deyu Zhou. 2024b. Denoising rationalization for multi-hop fact verification via multi-granular explainer. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12593–12608.

Jianheng Tang, Qifan Zhang, Yuhan Li, and Jia Li. 2024. Grapharena: Benchmarking large language models on graph computational problems. *arXiv preprint arXiv:2407.00379*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. *arXiv preprint arXiv:2408.04840*.

Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. 2024a. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*.

Xin Zhang, Linhai Zhang, and Deyu Zhou. 2023. Sentiment analysis on streaming user reviews via dual-channel dynamic graph neural network. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7208–7220.

Yu Zhang, Kehai Chen, Xuefeng Bai, Zhao Kang, Quanjiang Guo, and Min Zhang. 2024b. Question-guided knowledge graph re-scoring and injection for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8972–8985.

Yu Zhang, Jinlong Ma, Yongshuai Hou, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. 2025. Evaluating and steering modality preferences in multimodal large language model. *arXiv preprint arXiv:2505.20977*.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124.

Xiangru Zhu, Zhixu Li, Xiaodan Wang, Xueyao Jiang, Penglei Sun, Xuwu Wang, Yanghua Xiao, and Nicholas Jing Yuan. 2024. Multi-modal knowledge graph construction and application: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):715–735.

A VGCure Construction

A.1 Graphs Generation

For synthetic graph structures, we use the NetworkX library for random generation and employ hyperparameters to control the expected macroscopic properties of each graph:

- **ER**: This structure takes an edge probablity parameter p, which we choose randomly from $\{0.2, 0.3, 0.4\}$ during generation.
- **BA**: This structure takes the parameter m, which denotes the number of edges to attach from a new node to existing nodes. We choose randomly from $\{2, 3\}$ during generation.
- **SFN**: For this structure, we use the default parameters provided by NetworkX except for the number of nodes during generation.
- **SBM**: This structure takes the sizes of blocks s and the density of edges going from the nodes of one group to nodes of another group p as parameters. During generation, we set s to $[n, m]$ and p to $[[p_1, p_2], [p_2, p_3]]$, where n and m are a random integer from $[3, 7]$ and $[4, 8]$, respectively, and p_1, p_2, p_3 are all randomly selected from $\{0.2, 0.3, 0.4\}$.
- **Star**: This structure requires no parameters other than the number of nodes.

For all the above structures except SBM, the number of nodes during generation is an arbitrary integer in the range $[7, 15]$.

To anonymize the visual graph, we use the unique name Ex containing no information to name the nodes in the graph structure, where $x \in \{1, 2, \ldots, n\}$ and n is the number of nodes in the graph. For edges, we choose a random identify

from {R1, R2, ... , R10} to name them. The name can be repeated for each edge. The examples of synthetic visual graph are shown in Fig.7.

A.2 Tasks Generation

To ensure the correctness of the generated samples, we first search for relevant paths in the given graph that satisfy the conditions of the task. Then the final QA samples and FC samples are generated based on the paths and corresponding templates. If no path exists, the generation of samples for the task is skipped. The final statistics of VGCURE and the example samples with undirected graph for each task are shown in Tab.9 and Tab.10, respectively.

B Experimental Setup for Evaluation

B.1 Prompts

To facilitate the LVLMs to understand the content in the visual graph, we take a **visual graph description** in addition to the input during test.

- **Directed Visual Graph Description**: The given image shows a graph where circles represent nodes, with the content inside indicating the node names. The arrowed lines connecting two nodes represent edges, and the content in the middle of the edges indicates the edge names.
- **Undirected Visual Graph Description**: The given image shows a graph where circles represent nodes, with the content inside being the node names. The lines connecting two nodes represent edges, and the content in the middle of the edges represents the edge names.

The complete prompt for QA samples are as follow:

- **NN, CR, RA, SRN, NQ**: [Visual Graph Description] Answer the given questions based on the graph in the image.\nQuestion: [question]\nPlease provide the answer directly without the reasoning process and present your answer in the LIST format: [Entity1, Entity2, ...].
- **NR**: [Visual Graph Description] Answer the given questions based on the graph in the image.\nQuestion: [question]\nPlease provide the answer directly without the reasoning process and present your answer in the LIST format: [Relation1, Relation2, ...].
- **CP**: [Visual Graph Description] Answer the given questions based on the graph in the image.\nQuestion: [question]\nIf yes, please output all the shortest paths in the LIST Format and conclude your answer with "Yes. The shortest paths are [[Entity1, Entity2,...], [Entity3, Entity4,...], ...]". If no path exists, please answer "No".
- **NNu, EN, DQ**: [Visual Graph Description] Answer the given questions based on the graph in the image.\nQuestion: [question]\nPlease provide the answer directly without the reasoning process.

The complete prompt for FC samples are as follow:

- [Visual Graph Description] Verify the truth of the given claim against the graph in the image.\nClaim: [claim]\nThe answer should contain only "True" or "False", without reasoning process.

B.2 Evaluation Metrics

For the QA samples of NQ, NN, CR, RA, SRN, and NR tasks, we use (macro-averaged) F1 score and Hits@1 as in the previous QA benchmarks (Rajpurkar et al., 2016; Zhang et al., 2018). For the QA samples of CP task, we employ EM_F1, which is the macro F1 score calculated based on the exact match between the predicted path and the ground truth path, and Label_Acc, which measures the accuracy of the model's prediction on whether a path exists or not. For the QA samples of NNu, EN and DQ tasks, we compute the accuracy between predicted answers and ground truth. For the FC samples of all tasks, following Si et al. (2024a,b), we use macro F1 and accuracy as the metrics.

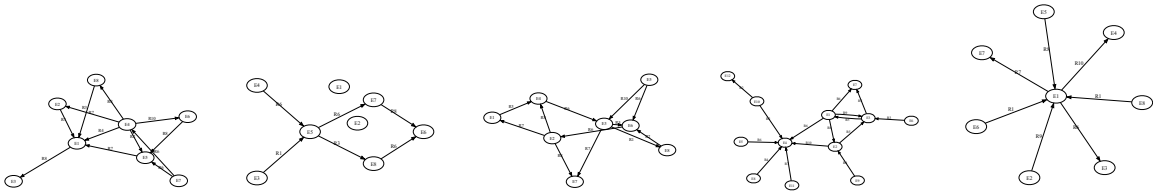
C Why LVLMs Fail on Fundamental Tasks?

C.1 Comparison with Backbone LLMs

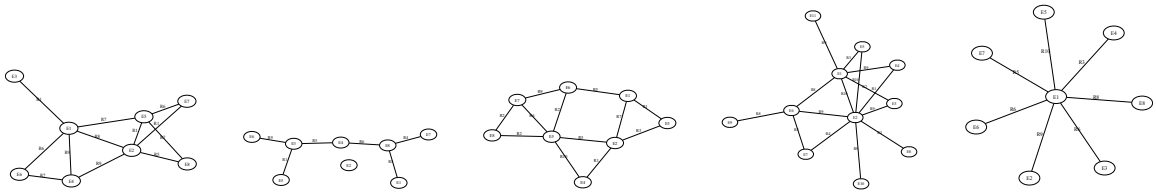
To compare the performance of the LVLMs with the corresponding LLMs on the VGCURE task, we first randomly select 1000 samples for each task and convert the corresponding visual graphs into text triples to construct the text version of the samples, and then evaluate them with the corresponding backbone LLMs.

The complete prompts for QA samples are as follows:

- **NN, CR, RA, SRN, NQ**: Given a set of triples representing an directed/undirected graph, where each triple denotes [Node, Edge, Node], answer the given questions based on the graph.\nTriples: [Triples of visual graph]\nQuestion: [question]\nPlease provide

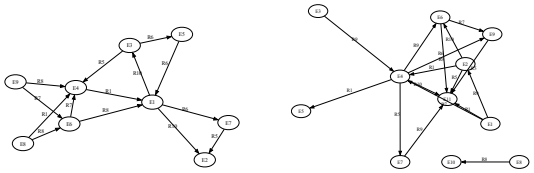


(a) Directed BA (b) Directed ER (c) Directed SBM (d) Directed SFN (e) Directed Star

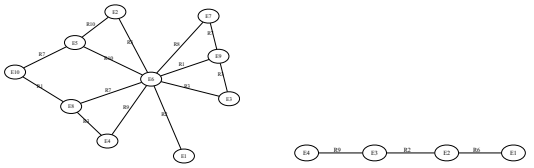


(f) Undirected BA (g) Undirected ER (h) Undirected SBM (i) Undirected SFN (j) Undirected Star

Figure 7: Examples of synthetic visual graphs.



(a) (b)



(c) (d)

Figure 8: Visual graphs corresponding to the examples of error type.

the answer directly without the reasoning process and present your answer in the LIST format: [Entity1, Entity2, ...].

- **NR**: Given a set of triples representing an directed/undirected graph, where each triple denotes [Node, Edge, Node], answer the given questions based on the graph.\nTriples: [Triples of visual graph]\nQuestion: [question]\nPlease provide the answer directly without the reasoning process and present your answer in the LIST format: [Relation1, Relation2, ...].
- **NNu, EN, DQ**: Given a set of triples representing an directed/undirected graph, where each triple denotes [Node, Edge, Node], answer the given questions based on the graph.\nTriples: [Triples of visual graph]\nQuestion: [question]\nPlease provide the answer directly without the reasoning process.

The complete prompt for FC samples are as follow:

- [Visual Graph Description] Given a set of triples representing an directed graph, where each triple denotes [Node, Edge, Node], verify the truth of the given claim against the graph.\nTriples: [Triples of visual graph]\nClaim: [claim]\nThe answer should contain only "True" or "False", without reasoning process.

The *Connective Path Query (CP)* task is ignored here due to LLMs' poor instruction-following ability on this task.

D Experimental Setup for Training

D.1 Training Samples

Graphs Generation For Masked Graph Infilling and Graph Description task, we use the same synthetic visual graph generation strategy as VGCURE. As for the Contrastive Graph Discrimination, in order to reduce the difficulty, we limited the number of nodes per graph structure to $[4, 8]$ during generation.

Task Instruction To increase the diversity of samples, we designed various instructions with similar semantics for each task in the MCDGRAPH.

- **Masked Graph Infilling**
  - Using the given set of triples, predict the word that should fill the [MASK] position in the knowledge graph in the image.

Structue Type		# Graphs	Number of QA Samples												Avg. Nodes	Avg. Edges
			NN	CR	CN	RA	SRN	NR	CP	NNu	EN	DC	DQ	NQ		
BA	Directed	400	400	400	400	370	324	400	400	400	400	400	400	400	10.98	20.99
	Undirected	400	400	400	400	400	400	400	400	400	400	400	400	400	11.02	21.04
ER	Directed	400	387	377	377	304	229	387	400	400	400	400	400	400	11.06	17.69
	Undirected	400	396	395	396	346	346	395	400	400	400	400	400	400	11.00	17.41
SBM	Directed	400	399	392	392	314	399	400	253	400	400	400	400	400	10.99	17.13
	Undirected	400	399	399	399	399	400	371	371	400	400	400	400	400	11.04	17.32
SFN	Directed	400	400	400	400	331	149	400	400	400	400	400	400	400	10.95	13.86
	Undirected	400	400	400	400	400	400	379	379	400	400	400	400	400	11.05	12.80
Star	Directed	400	396	388	388	350	288	396	400	400	400	400	400	400	12.05	11.05
	Undirected	400	400	400	400	393	393	400	400	400	400	400	400	400	12.11	11.11
DBLP	Directed	200	196	192	192	140	196	200	145	200	200	200	200	200	7.85	17.76
	Undirected	200	200	200	200	173	173	200	200	200	200	200	200	200	7.85	17.76
Dbpedia	Directed	200	186	185	185	93	186	200	101	200	200	200	200	200	8.13	11.36
	Undirected	200	200	200	200	155	155	200	200	200	200	200	200	200	8.13	11.36
Openflights	Directed	100	100	100	100	65	100	100	63	100	100	100	100	100	5.51	13.59
	Undirected	100	100	100	100	90	90	100	100	100	100	100	100	100	5.51	13.59
PubChemQC	Directed	400	400	138	400	400	119	119	37	400	400	400	400	400	5.45	4.76
	Undirected	400	400	398	400	398	400	151	151	400	400	400	400	400	5.45	4.76
Social Network	Directed	300	262	254	254	134	262	300	124	300	300	300	300	300	7.57	9.25
	Undirected	300	299	297	299	238	238	297	300	300	300	300	300	300	7.57	9.25
Total		6400	6320	6015	6282	5493	5247	5795	5224	6400	6400	6400	6400	6400	-	-

Table 9: Statistics of VGCURE benchmark, where *# Graphs* represents the number of visual graphs, *Avg.Nodes* and *Avg.Edges* denote the average number of nodes and edges in the graph, respectively. For each task, the number of QA samples is the values in the table, except for CN and DC, which have no QA samples, and the number of FC samples is **twice** the value in the table.

- Based on the provided triples, determine the correct word to complete the [MASK] position in the knowledge graph shown in the image.
- Given the set of triples, predict the word that should be placed in the [MASK] position within the knowledge graph in the image.
- Use the given triples to predict the appropriate word for the [MASK] position in the knowledge graph depicted in the image.
- Using the set of triples, identify the word that should fill the [MASK] position in the knowledge graph in the image.
- Based on the set of triples, predict the correct word for the [MASK] position in the knowledge graph in the image.
- Given the triples, predict the word that fits the [MASK] position in the knowledge graph present in the image.
- Using the triples provided, determine the word that should be used to fill the [MASK] position in the knowledge graph in the image.
- Predict the word that should occupy the [MASK] position in the knowledge graph in the image, based on the given triples.
- Using the provided triples, identify the word that should complete the [MASK] position in the knowledge graph in the image.

• **Contrastive Graph Discrimination**

- Determine whether Graph A and Graph B in the given image are identical.
- Assess if Graph A and Graph B depicted in the image are equivalent.
- Evaluate whether the structures of Graph A and Graph B in the provided image match.
- Identify if there are any differences between Graph A and Graph B in the shown image.
- Check if Graph A and Graph B illustrated in the image are the same.
- Analyze the image to determine if Graph A is identical to Graph B.
- Investigate whether Graph A and Graph B in the given image are congruent.
- Examine the provided image to see if Graph A and Graph B are equivalent.
- Compare Graph A and Graph B in the image to establish their similarity.
- Confirm if Graph A and Graph B presented in

Task	QA sample	FC sample (Label)
NNu	Q: How many nodes are there in this graph? A: 11	There are 11 nodes in this graph. (True) There are 15 nodes in this graph. (False)
EN	Q: How many edges are there in this graph? A: 15	There are 15 edges in this graph. (True) There are 19 edges in this graph. (True)
DC	-	This graph is an undirected graph. (True) This graph is a directed graph. (True)
DQ	Q: What is the degree of E7 in this graph? A: 2	The degree of E7 in this graph is 2. (True) The degree of E7 in this graph is 7. (False)
NQ	Q: Which nodes are neighbors of E6 in this graph? A: [E1, E2, E7, E9]	E7 is a neighbors of E6 in this graph. (True) E10 is a neighbors of E6 in this graph. (False)
NN	Q: Which entities are connected to the entity that has R10 with E2 via R9? A: [E11, E4]	E4 is connected to the entity that has R10 with E2 via R9. (True) E3 is connected to the entity that has R10 with E2 via R9. (False)
CR	Q: Which entities are connected to E8 via R3 as well as connected E1 via R10? A: [E2]	E2 is connected to E8 via R3 as well as connected E1 via R10. (True) E3 is connected to E8 via R3 as well as connected E1 via R10. (False)
CN	-	E5 and E2 share a common neighbor. (True) E10 and E11 share a common neighbor. (False)
RA	Q: Which entities are connected to E1 via the same relation between E11 and E1? A: [E4]	E4 is connected to E1 via the same relation between E11 and E1. (True) E2 is connected to E1 via the same relation between E11 and E1. (False)
SRN	Q: Which entities are both connected to E2 via R10? A: [E1, E5]	E5 and E1 both connected to E2 via R10. (True) E5 and E3 are both connected to E2 via R10. (False)
NR	Q: What is the relation between E2 and the entity that is connected to E6 via R8? A: [R10]	The relation between E2 and the entity that is connected to E6 via R8 is R10. (True) The relation between E2 and the entity that is connected to E6 via R8 is R4. (False)
CP	Q: Is there a path between E5 and E3? A: Yes. The shortest paths are [[E5, E1, E3], [E5, E2, E3]]	[E5, E1, E3] is one of the shortest path between E5 and E3. (True) [E5, E11, E2, E3] is one of the shortest path between E5 and E3. (False)

Table 10: Examples with undirected graph for each task in VGCURE. These samples all correspond to the SFN graph shown in Fig.7(i).

the image are indistinguishable.

- **Graph Description**
  - Describe the given graph, including the number of nodes, the number of edges, and the names of all the nodes.
  - Provide a description of the given graph, specifying the number of nodes, edges, and listing all the node names.
  - Analyze the given graph by stating the number of nodes, edges, and enumerating the names of all the nodes.
  - Summarize the graph by detailing the number of nodes, edges, and listing the names of each node.
  - Explain the graph, including the count of nodes and edges, and provide the names of all the nodes.
  - Describe the graph, indicating how many nodes and edges it contains, and listing all the node names.
  - Provide an overview of the graph, mentioning the number of nodes, edges, and the names of all nodes.
  - Characterize the given graph, noting the number of nodes, edges, and listing all the node names.
  - Detail the structure of the given graph, including node and edge counts, and providing a list of all node names.
  - Give a description of the graph, including the total number of nodes, edges, and the names of all the nodes.

Similar to VGCURE, we include a visual graph description in input as well. Thus, the complete **task instruction** I for each training sample in MCDGRAPH is "[Visual Graph Description] [Instruction]".

Number of Samples For Masked Graph Infilling task, we generate **10,000 samples**, with half of the samples masking nodes and the other half masking edges. For Contrastive Graph Discrimination tasks, **5,000 samples**, where each sample consists of two visual graphs, are generated automatically. Similarly, the Graph Description task also contains **5,000 samples**, each corresponding to a unique visual graphs.

D.2 Implementation Details

For Qwen2-VL, we employ the LoRA-based supervised fine-tuning scripts provided by LLaMA-Factory[3]. For InternVL2, we perform LoRA-based fine-tuning based on the code and documentation provided officially[4]. The hyperparameters used for training are shown in Tab.12. All the experiments are finished on 4 A100 GPUs with 80GB memory.

[3] https://github.com/hiyouga/LLaMA-Factory
[4] https://github.com/OpenGVLab/InternVL

Error Type	QA samples	Visual Graph
Relation Misunderstanding	Q: What is the relation from the entity that is R5 of E3 to E1? A: [R1] P: [R6]	Fig.8(a)
Complexity Limitation	Q: What is the relation from the entity that is R7 of E1 to E7? A: [R5] P: [R9]	Fig.8(b)
Structural Hallucination	Q: Is there a path between E1 and E3? A: Yes. The shortest paths are [[E1, E6, E3]] P: Yes. The shortest paths are [E1, E3] and [E1, E4, E3].	Fig.8(c)
Entity-based Answering	Q: Which entities are R5 of E2 as well as R1 of E1? A: [E11] P: [E2, E1]	Fig.8(b)
Off-target Generation	Q: What is the relation between E2 and the entity that is connected to E4 via R9? A: [R2] P: [Edge, Node]	Fig.8(d)
Format Error	Q: Is there a path between E4 and E1? A: Yes. The shortest paths are [[E4, E3, E2, E1]] P: Yes. The shortest paths are [[E4, R9, E3, R2, E2, R6, E1]].	Fig.8(d)

Table 11: Examples of each error type, where *Q* denotes question, *A* denotes gold answer and *P* denotes prediction.

Model	Lora_rank	Lora_ alpha	Global Batch Size	Learning rate	Epoch
Qwen2-VL	8	16	64	1e-4	5
InternVL2	64	128	64	4e-5	1

Table 12: Hyperparameters for training

Models	ScienceQA	AOKVQA
Qwen2-VL	85.13	84.37
w MCDGRAPH	81.31	83.67
InternVL2	97.07	85.41
w MCDGRAPH	96.88	84.45

Table 13: Model performance (Acc) on general VQA tasks.

E Generalization of MCDGRAPH

E.1 Impact on Visual Styles

To validate the generalization of our method, we employ *NetworkX* and *Matplotlib* to regenerate 50 visual graphs with different visual styles from those in VGCURE for each graph structure to explore the impact of visual graph styles. The examples of **the same** visual graph with different styles are illustrated in Fig.10 and Fig.11.

As the results shown in Tabs.14 and 15, although LVLMs never encounter the different style of visual graph during fine-tuning, our method can still improve the performance of the LVLMs on almost all tasks. This demonstrates the ability of our method to enhance LVLMs' ability in capturing the structural information in visual graphs with excellent generalization. In addition, compared to Tab.6, it can be noticed that the experimental results before and after the change of style are similar, which indicates that the style itself has no effect on the evaluation of LVLMs.

E.2 Impact of Naming Conventions

To explore the impact of altering the naming conventions, we regenerated the visual graphs and samples using the following new naming conventions.

- **Node-Edge**: The nodes in the visual graph are renamed **Node**x, where $x \in \{1, 2, \ldots, n\}$ and n is the number of nodes in the graph, and the edges are renamed **Edge**y, where $y \in \{1, 2, \ldots, 10\}$.
- **Name-R**: The nodes in the visual graph are renamed to a simple and common **human name** without any semantic bias, like "John", "Jane", "Mike", "Mary", etc. The names of the edges remain as they are, i.e., R1, R2, ..., R10.

The results are shown in Tabs.16 and 17. We can observe that after altering the naming conventions, LVLMs continue to show similar trends on most of tasks. Therefore, the experimental analysis in the main text still holds. Meanwhile, the performance of LVLMs decreases on most of the tasks when confronted with different names. This might be due to the fact that the new naming conventions gives longer names to nodes and edges and recognizing these information increases the difficulty of the task. It is worth noting that in the face of the new naming conventions, our MCDGRAPH still improves the performance of LVLMs on most tasks, although the fine-tuning still uses the original

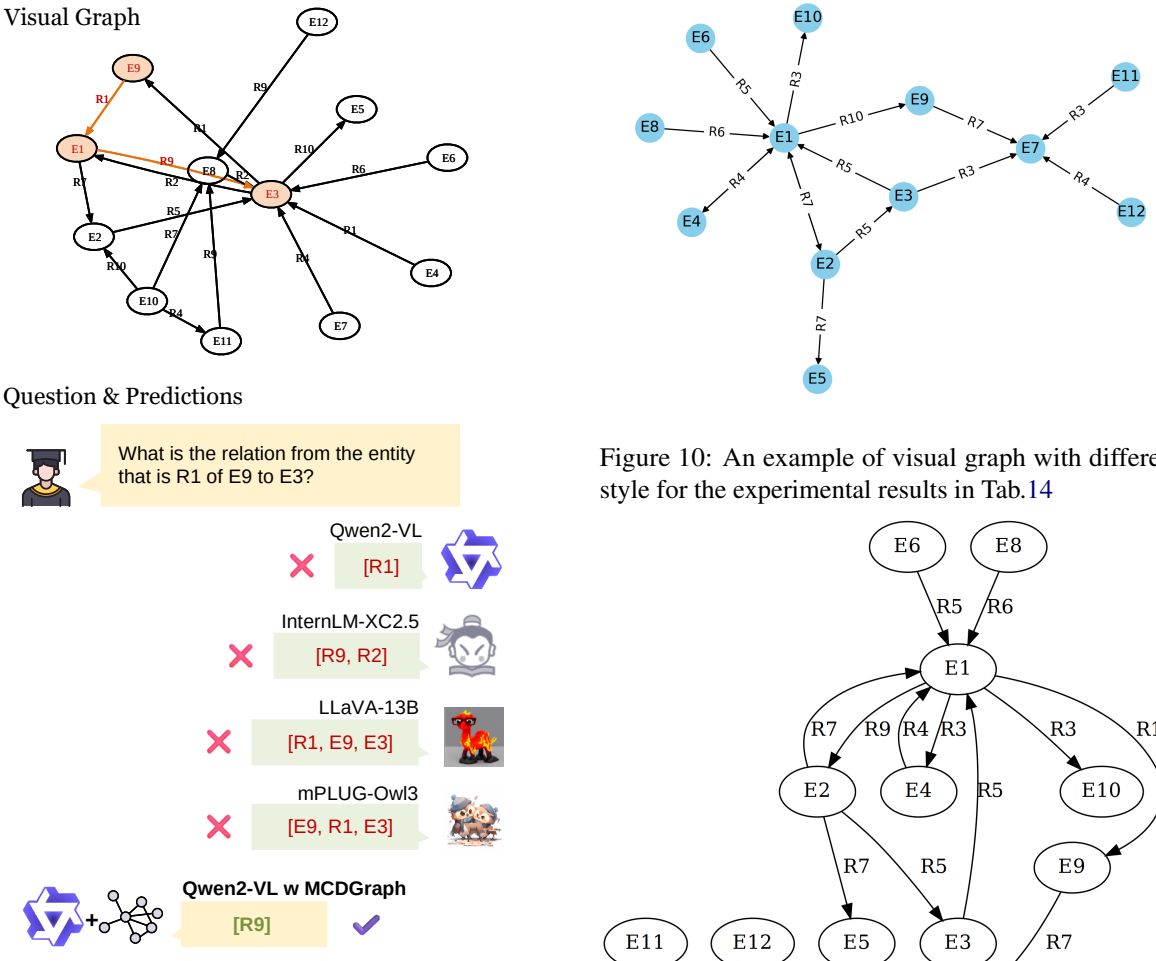


Figure 9: A case of Nested Relation Query task.

naming conventions. This strongly demonstrates the effectiveness as well as the generalization of our method.

F Results on General VQA Tasks

We evaluate our MCDGRAPH on two general VQA tasks, i.e., ScienceQA (Lu et al., 2022) and AOKVQA (Schwenk et al., 2022). According to the results shown in Tab.13, we can observe that after MCDGraph, LVLMs perform worse on these two VQA tasks. This is due to the fact that our MCDGRAPH is proposed for visual graph understanding and reasoning tasks, which are very different from general VQA tasks. And it is worth noting that our method does not overly compromise the LVLMs' performance on general VQA tasks, which we consider acceptable.

G Case Study

Fig.9 illustrates a case on *Nested Relation Query* task. We can observe that Qwen2-VL, LLaVA-13B and mPLUG-Owl3 all make the error of **Entity-based Answering** when confronted with this question, i.e., using "R1", "E9", and "E3" mentioned in the question as the generated answer. InternLM-XC2.5 makes the error of **Relation Misunderstanding**, i.e., the edge pointing from E3 to E1 is also used as the answer. However, Qwen2-VL after applying MCDGRAPH can answer this question correctly. This demonstrates that the proposed method can improve the fundamental graph structure understanding of LVLMs, thus avoiding the occurrence of the previously mentioned errors.

Figure 10: An example of visual graph with different style for the experimental results in Tab.14

Figure 11: An example of visual graph with different style for the experimental results in Tab.15

Models	Understanding					Reasoning						
	NNu	EN	DC	DQ	NQ	NN	CR	CN	RA	SRN	NR	CP
	QA Samples											
Qwen2-VL	98.20	18.10	-	50.70	18.64	15.66	22.66	-	14.83	20.45	16.95	11.04
w MCDGRAPH	99.40 ↑	40.20 ↑	-	56.80 ↑	28.44 ↑	14.64	27.02	-	13.69	22.22 ↑	14.73	11.99 ↑
InternVL2	73.70	8.60	-	48.40	26.33	18.26	26.19	-	9.85	21.75	11.20	1.60
w MCDGRAPH	95.80 ↑	35.70 ↑	-	52.90 ↑	28.89 ↑	19.90 ↑	29.24 ↑	-	11.73 ↑	23.23 ↑	18.04 ↑	↑ 14.49
	FC Samples											
Qwen2-VL	71.37	61.78	86.53	73.82	73.03	48.00	77.35	39.98	36.04	60.22	45.31	42.93
w MCDGRAPH	88.45 ↑	63.14 ↑	94.43 ↑	76.26 ↑	78.59 ↑	63.17 ↑	82.63 ↑	53.78 ↑	54.33 ↑	66.26 ↑	65.95 ↑	56.30 ↑
InternVL2	75.43	34.80	86.85	63.68	74.52	61.12	74.23	47.22	33.48	55.85	55.32	53.15
w MCDGRAPH	83.82 ↑	70.14 ↑	83.48	60.77	79.69 ↑	71.39 ↑	78.30 ↑	52.38 ↑	51.24 ↑	59.83 ↑	68.58 ↑	44.73

Table 14: Model performance (Acc/F1/EM_F1 for QA and F1 for FC) on various tasks with different visual styles. The example of the corresponding visual style is shown in Fig.10

Models	Understanding					Reasoning						
	NNu	EN	DC	DQ	NQ	NN	CR	CN	RA	SRN	NR	CP
	QA Samples											
Qwen2-VL	95.30	20.80	-	40.00	13.95	8.73	19.04	-	11.73	14.89	13.16	3.17
w MCDGRAPH	98.80 ↑	17.30	-	50.50 ↑	22.33 ↑	11.56 ↑	21.92 ↑	-	11.05	17.84 ↑	11.60	0.12
InternVL2	85.20	17.90	-	39.60	24.21	19.99	22.26	-	8.06	19.46	12.03	2.25
w MCDGRAPH	97.20 ↑	35.70 ↑	-	43.00 ↑	26.46 ↑	19.12	23.54 ↑	-	11.00 ↑	20.45 ↑	14.80 ↑	12.11 ↑
	FC Samples											
Qwen2-VL	70.88	68.06	66.94	57.90	65.09	43.62	67.32	34.28	34.79	53.23	42.54	42.89
w MCDGRAPH	76.36 ↑	58.47	85.20 ↑	74.38 ↑	73.69 ↑	56.97 ↑	75.05 ↑	48.68 ↑	53.84 ↑	60.99 ↑	60.44 ↑	56.93 ↑
InternVL2	68.42	38.35	91.84	64.58	65.45	62.91	72.42	48.17	34.60	57.88	57.77	52.48
w MCDGRAPH	69.45 ↑	69.11 ↑	84.10	62.09	74.68 ↑	71.52 ↑	75.19 ↑	52.97 ↑	50.89 ↑	61.00 ↑	66.44 ↑	43.19

Table 15: Model performance (Acc/F1/EM_F1 for QA and F1 for FC) on various tasks with different visual styles. The example of the corresponding visual style is shown in Fig.11

Models	Understanding					Reasoning						
	NNu	EN	DC	DQ	NQ	NN	CR	CN	RA	SRN	NR	CP
	QA Samples											
Qwen2-VL	94.30	18.10	-	48.40	17.55	11.34	13.72	-	10.10	11.82	5.34	10.63
w MCDGRAPH	94.10	16.00	-	63.50 ↑	17.25	8.50	14.56 ↑	-	8.29	12.81 ↑	12.71 ↑	13.00 ↑
InternVL2	65.61	19.15	-	47.94	17.45	12.11	15.75	-	8.87	13.54	8.25	8.12
w MCDGRAPH	89.30 ↑	34.60 ↑	-	49.20 ↑	17.59 ↑	12.88 ↑	14.63	-	8.29	13.75 ↑	4.52	5.45
	FC Samples											
Qwen2-VL	74.18	66.99	91.90	58.77	58.34	37.99	59.80	34.25	34.51	45.99	35.39	39.40
w MCDGRAPH	85.24 ↑	66.42	90.93	81.77 ↑	77.30 ↑	56.79 ↑	76.47 ↑	53.06 ↑	52.85 ↑	59.45 ↑	58.51 ↑	58.72 ↑
InternVL2	72.82	41.92	96.19	65.34	73.36	61.51	77.15	52.21	34.79	55.23	57.50	53.17
w MCDGRAPH	73.86 ↑	72.70 ↑	94.28	59.99	78.23 ↑	69.02 ↑	74.32	52.12	54.71 ↑	57.41 ↑	68.51 ↑	44.20

Table 16: Model performance (Acc/F1/EM_F1 for QA and F1 for FC) on various tasks with **Node-Edge** naming convention.

Models	Understanding					Reasoning						
	NNu	EN	DC	DQ	NQ	NN	CR	CN	RA	SRN	NR	CP
	QA Samples											
Qwen2-VL	21.00	22.90	-	51.10	17.78	11.80	18.31	-	11.50	15.33	8.66	14.02
w MCDGRAPH	13.50	25.70 ↑	-	63.30 ↑	18.66 ↑	9.56	18.91 ↑	-	10.13	15.90 ↑	10.81 ↑	11.72
InternVL2	32.80	17.70	-	55.10	19.77	14.61	21.60	-	10.69	16.20	11.29	11.79
w MCDGRAPH	24.40	36.00 ↑	-	54.20	21.10 ↑	13.32	21.44	-	8.15	16.01	17.77 ↑	11.84 ↑
	FC Samples											
Qwen2-VL	86.35	67.03	93.73	69.39	74.05	41.30	71.54	43.59	38.54	58.05	42.43	44.18
w MCDGRAPH	68.68	69.10 ↑	96.45 ↑	78.20 ↑	76.96 ↑	57.61 ↑	82.66 ↑	56.07 ↑	55.19 ↑	63.51 ↑	63.17 ↑	56.45 ↑
InternVL2	69.34	39.80	87.75	62.98	76.17	69.06	72.14	57.15	39.82	56.87	65.83	48.31
w MCDGRAPH	78.05 ↑	71.65 ↑	75.85	55.84	81.23 ↑	71.50 ↑	79.29 ↑	52.43	57.76 ↑	53.85	67.60 ↑	41.99

Table 17: Model performance (Acc/F1/EM_F1 for QA and F1 for FC) on various tasks with **Name-R** naming convention.

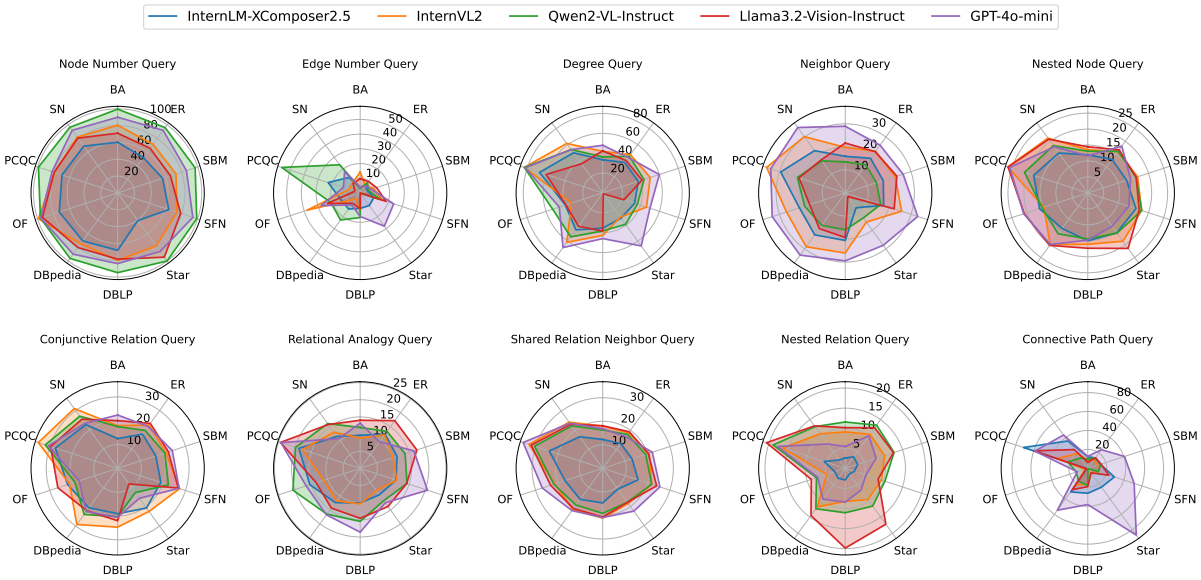


Figure 12: Model performance (Acc/F1) on QA samples across various graph structures and tasks.

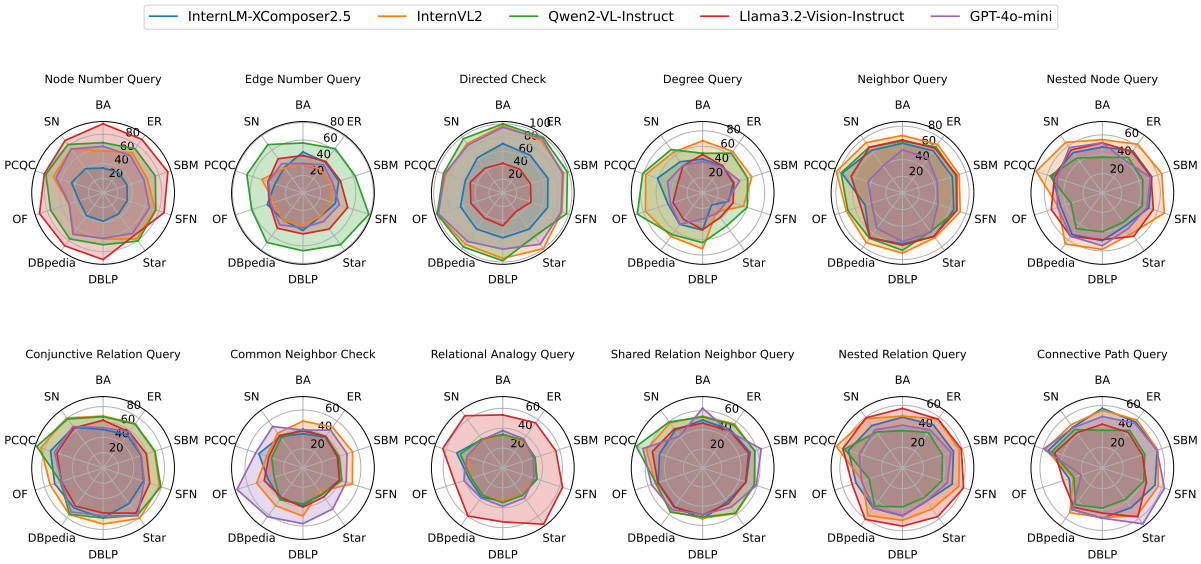


Figure 13: Model performance (F1) on FC samples across various graph structures and tasks.

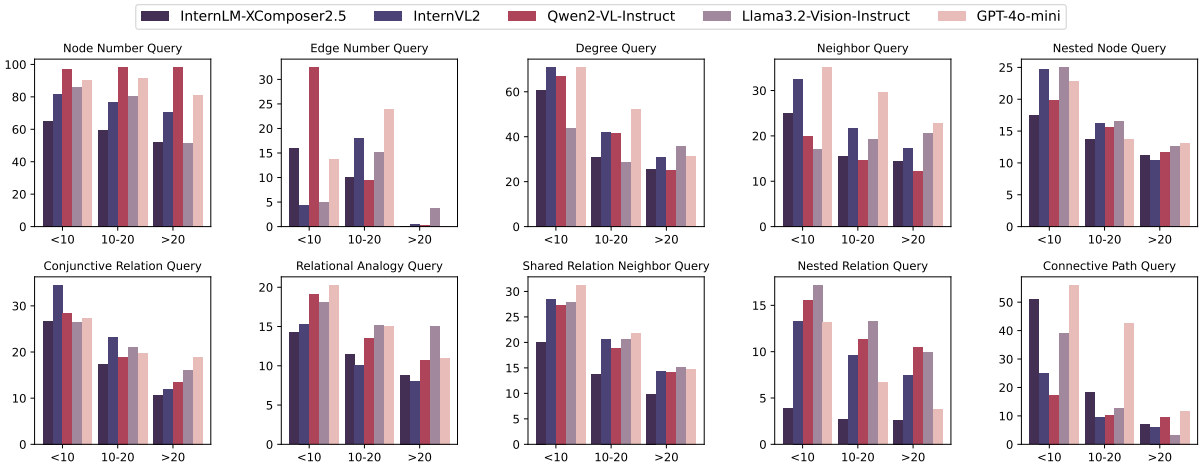


Figure 14: Model performance (F1/Acc) comparison on QA samples across various tasks and **edge** ranges.

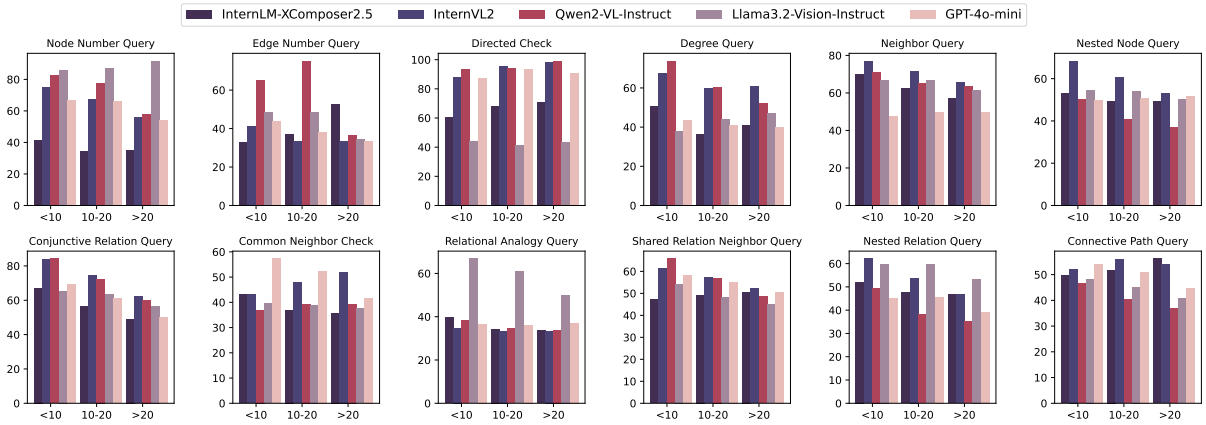


Figure 15: Model performance (F1) comparison on FC samples across various tasks and **edge** ranges.

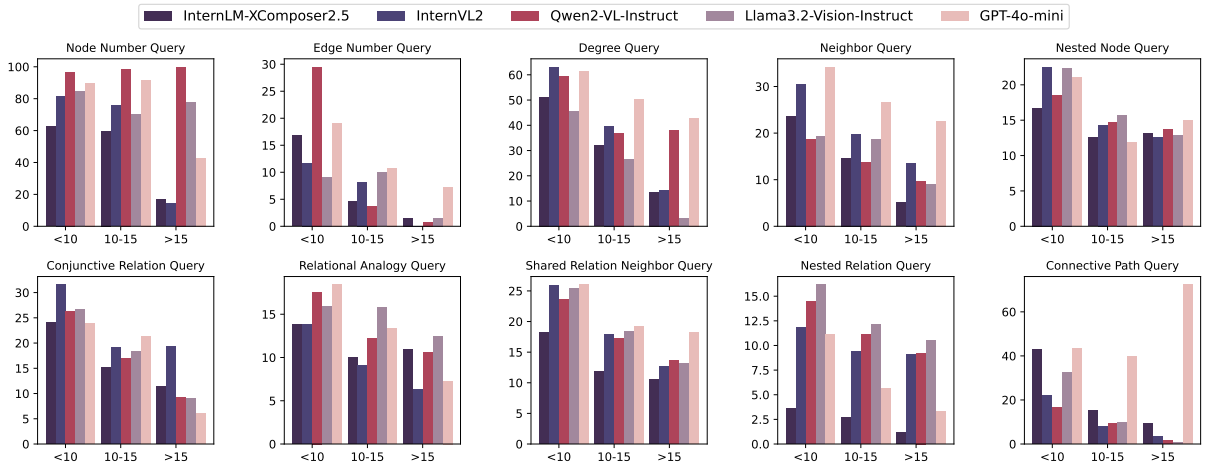


Figure 16: Model performance (F1/Acc) comparison on QA samples across various tasks and **node** ranges.

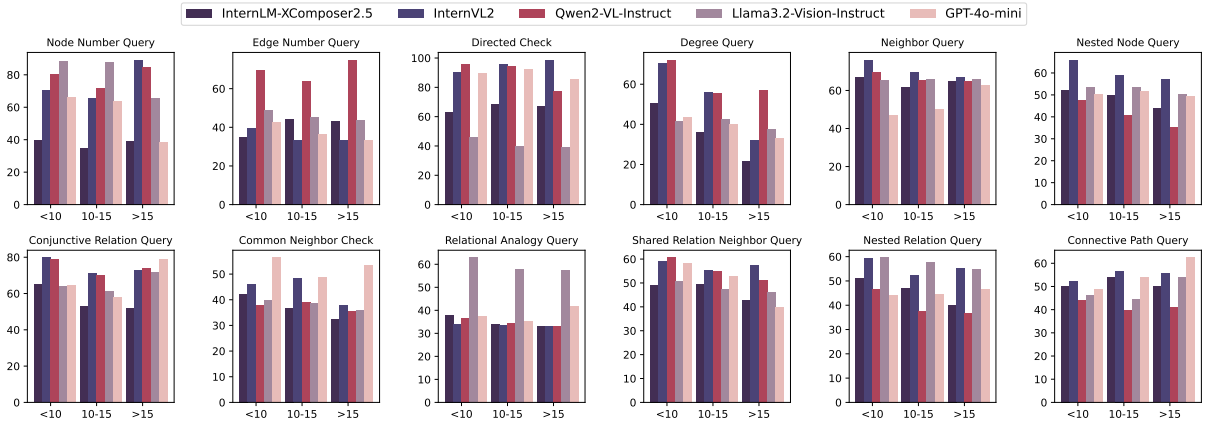


Figure 17: Model performance (F1) comparison on FC samples across various tasks and **node** ranges.

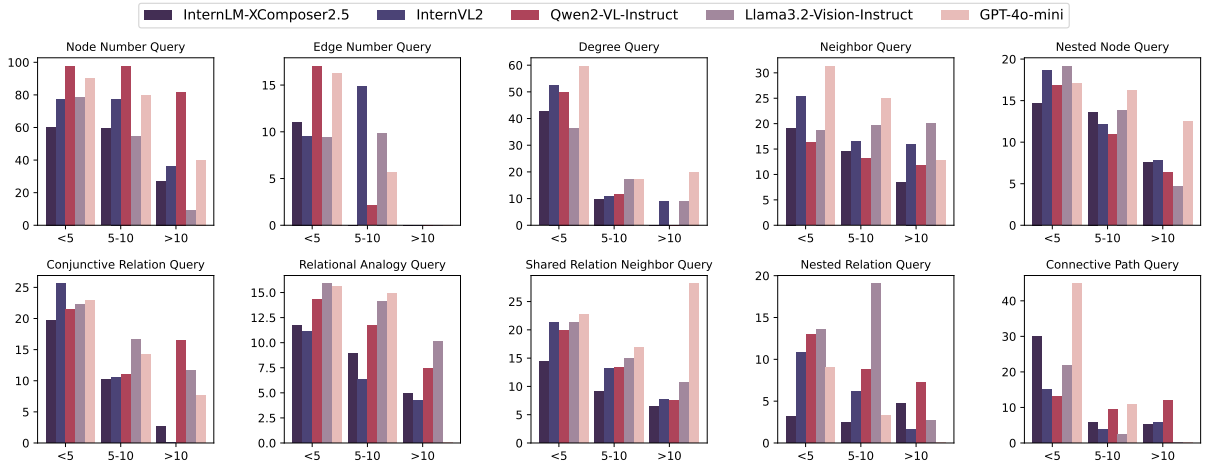


Figure 18: Model performance (F1/Acc) comparison on QA samples across various tasks and **average degree**.

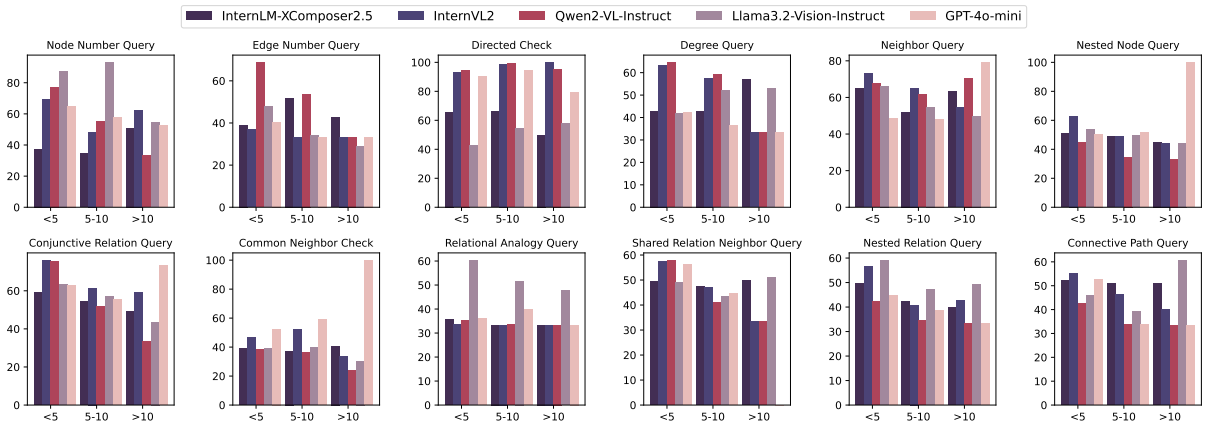


Figure 19: Model performance (F1) comparison on FC samples across various tasks and **average degree**.

Models	QA Samples																	
	Understanding					Reasoning												
	NNu	EN	DQ	NQ		NN		CR		RA		SRN		NR		CP		
	Acc	Acc	Acc	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1	F1	Hits@1	EM_F1	Label_Acc
Qwen2-VL	97.80	16.38	48.09	16.18	38.12	16.52	28.34	21.02	48.57	14.19	27.96	19.48	56.42	12.73	25.07	12.90	38.06
w MCDGRAPH	98.34 ↑	25.92 ↑	60.94 ↑	25.44 ↑	63.44 ↑	13.32	28.97 ↑	26.14 ↑	63.84 ↑	13.14	28.42 ↑	20.74 ↑	62.21 ↑	14.44 ↑	23.71 ↑	11.95	63.98 ↑
InternVL2	77.45	9.78	50.75	25.01	68.58	18.30	30.82	24.87	59.31	10.83	17.12	20.72	59.99	10.58	18.97	14.53	43.97
w MCDGRAPH	95.68 ↑	40.45 ↑	54.78 ↑	28.80 ↑	72.72 ↑	19.43 ↑	27.86	28.53 ↑	67.61 ↑	11.67 ↑	19.81 ↑	22.34 ↑	61.67 ↑	16.50 ↑	40.21 ↑	12.76	25.23

Table 18: Performance Improvement of MCDGRAPH on QA samples across various tasks. ↑ indicates an improvement compared to the original model.

Models	FC Samples																					
	Understanding										Reasoning											
	NNu		EN		DC		DQ		NQ		NN		CR		CN		RA		SRN		NR		CP	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Qwen2-VL	76.50	77.67	68.26	68.27	94.92	94.94	64.32	67.40	67.34	68.76	44.17	53.77	74.28	75.24	38.52	51.10	35.28	50.55	57.07	59.18	42.08	52.95	42.25	53.38
w MCDGRAPH	89.58 ↑	89.69 ↑	65.80	68.64 ↑	95.84 ↑	95.84 ↑	77.10 ↑	77.23 ↑	79.75 ↑	80.03 ↑	60.71 ↑	63.35 ↑	83.07 ↑	83.08 ↑	53.90 ↑	54.12 ↑	53.48 ↑	53.69 ↑	64.17 ↑	64.21 ↑	64.12 ↑	65.51 ↑	60.11 ↑	60.17 ↑
InternVL2	68.63	71.18	36.82	50.23	93.17	93.18	63.28	63.84	72.81	73.27	62.46	63.62	75.37	76.18	47.12	50.40	33.70	50.14	56.98	57.42	55.94	58.42	54.71	54.71
w MCDGRAPH	76.55 ↑	77.71 ↑	71.98 ↑	72.02 ↑	90.04	90.50	56.83	58.53	80.98 ↑	81.89 ↑	73.14 ↑	73.98 ↑	80.23 ↑	80.69 ↑	52.09 ↑	52.99 ↑	52.81 ↑	57.08 ↑	59.07 ↑	60.07 ↑	69.03 ↑	70.24 ↑	45.82	49.27

Table 19: Performance Improvement of MCDGRAPH on FC samples across various tasks. ↑ indicates an improvement compared to the original model.

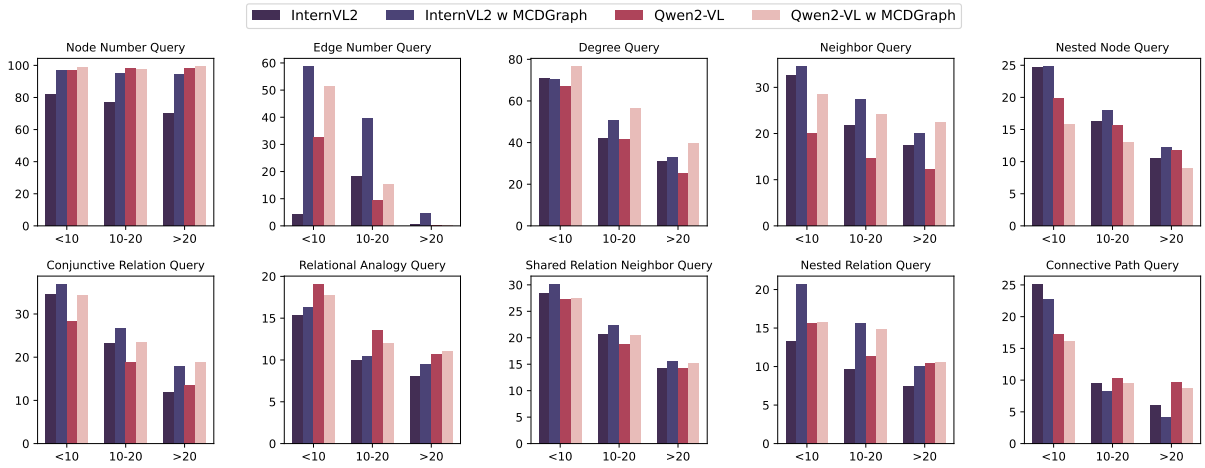


Figure 20: Performance Improvement of MCDGRAPH (F1/Acc) on QA samples across various tasks and **edge** ranges.

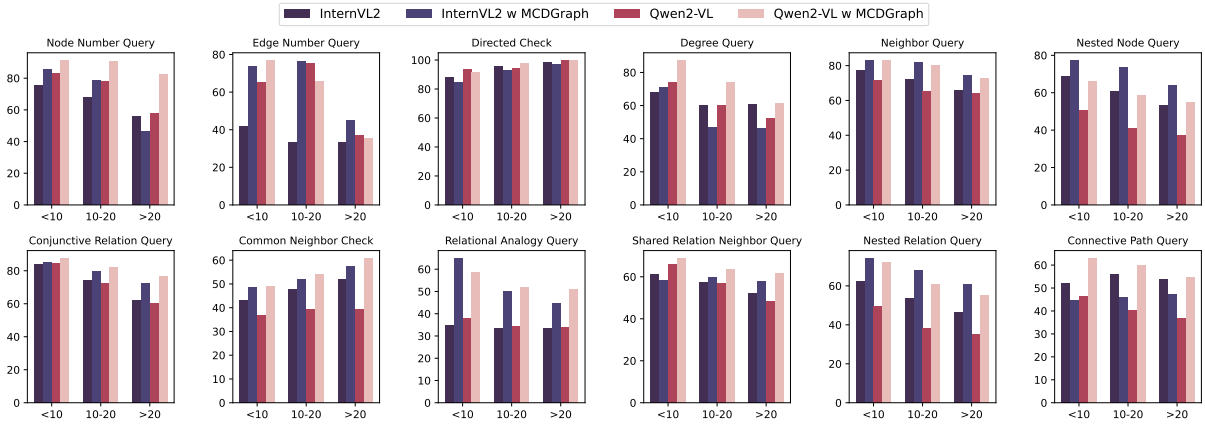


Figure 21: Performance Improvement of MCDGRAPH (F1) comparison on FC samples across various tasks and **edge** ranges.

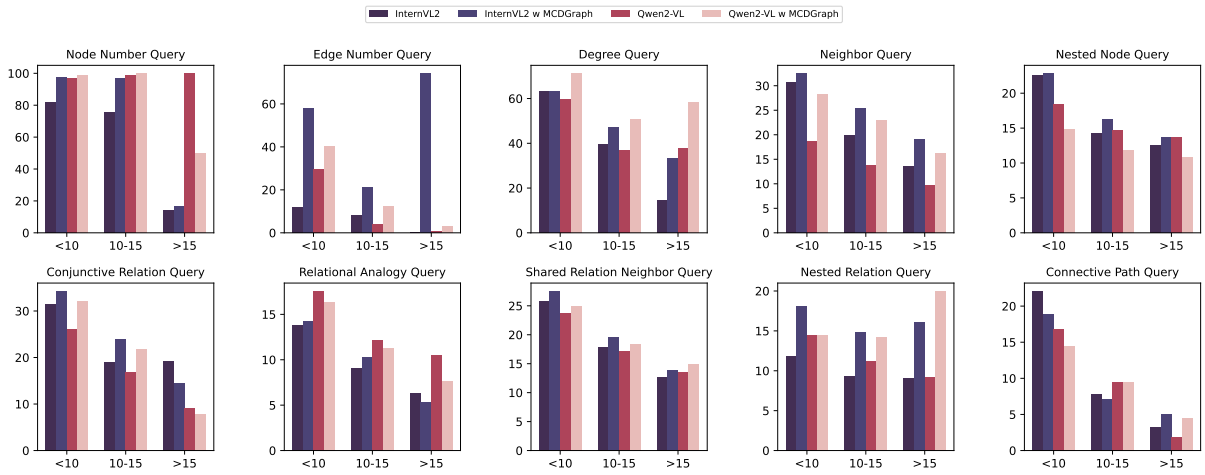


Figure 22: Performance Improvement of MCDGRAPH (F1/Acc) on QA samples across various tasks and **node** ranges.

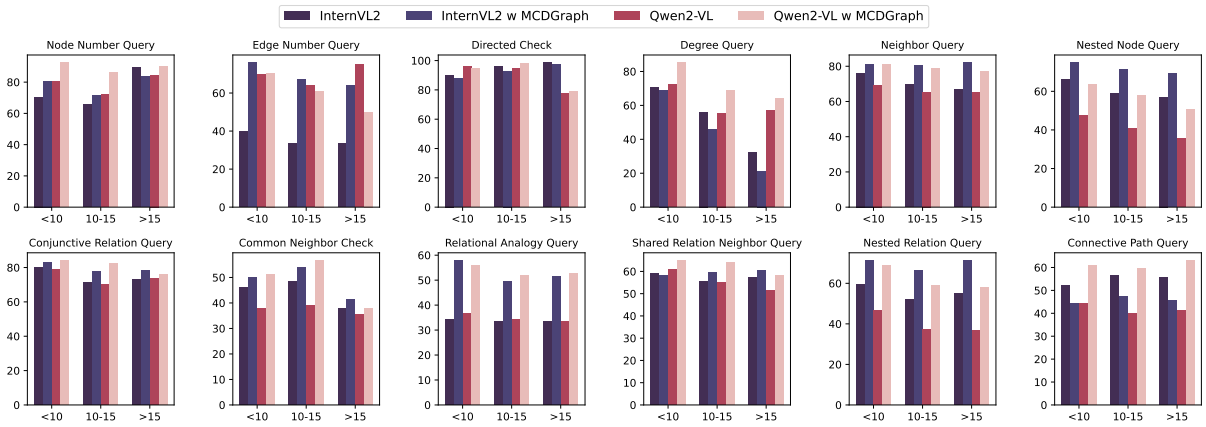


Figure 23: Performance Improvement of MCDGRAPH (F1) comparison on FC samples across various tasks and **node** ranges.

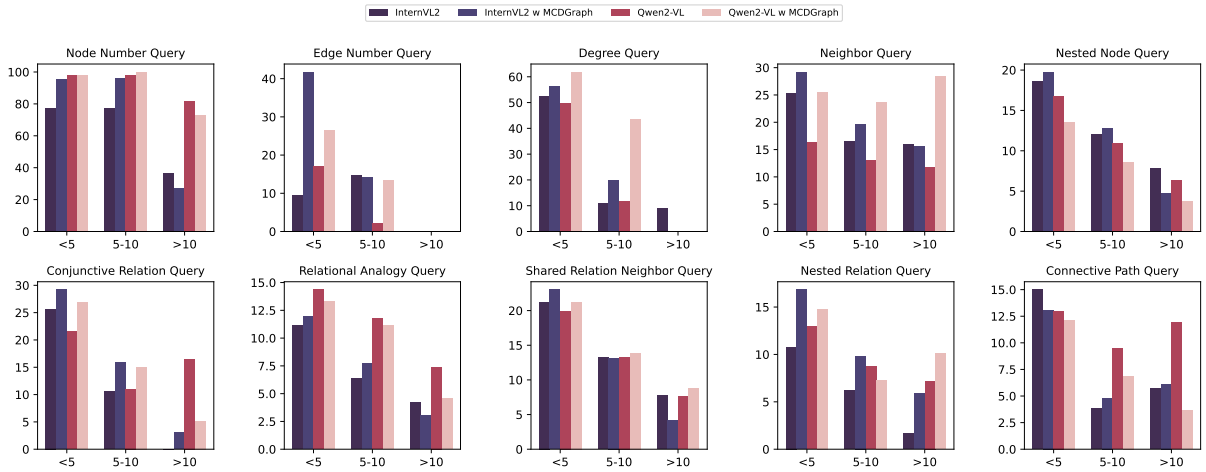


Figure 24: Performance Improvement of MCDGRAPH (F1/Acc) on QA samples across various tasks and **average degree**.

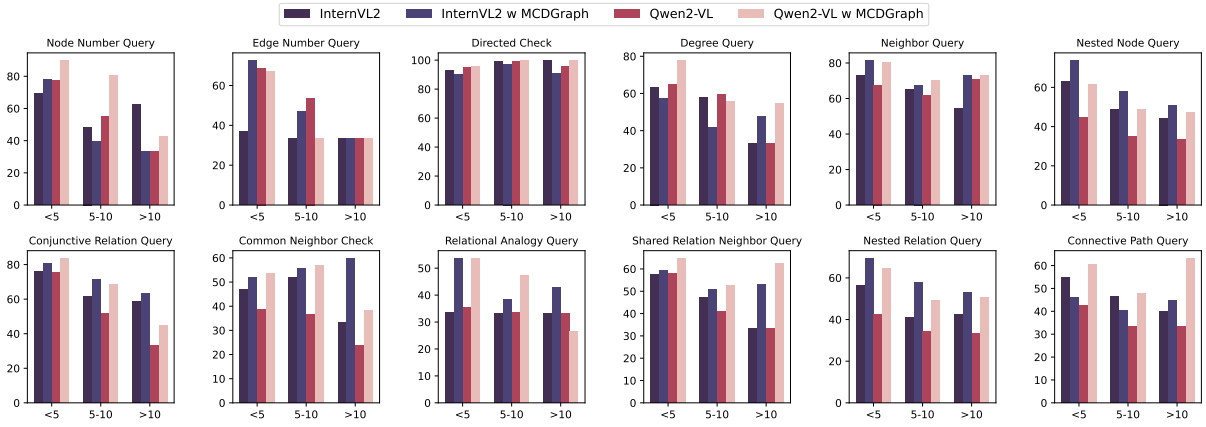


Figure 25: Performance Improvement of MCDGRAPH (F1) comparison on FC samples across various tasks and **average degree**.