VISUAL GRAPH HAR(D)NESS: FROM PERCEPTION TO REASONING IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Visual Graph Harness (VGH[1]), a comprehensive benchmark for evaluating graph understanding and reasoning in vision-language models (VLMs) from graph images. Unlike text-based graph benchmarks, VGH requires models to first perceive and interpret graph structure from visual input before performing graph computations. The benchmark covers a broad spectrum of capabilities, ranging from fine-grained graph perception tasks, such as determining node and edge existence, counting nodes and edges, to higher-level algorithmic reasoning tasks, including shortest-path computation, graph diameter, connected components, maximum clique, minimum vertex cover, maximum independent set, traveling salesman, and chordless cycle. VGH includes real-world graphs across diverse domains, including knowledge graphs, route maps, social networks, and commutative diagrams from topology. Commutative diagrams introduce additional visual challenges, such as mathematical notation, repeated node labels, and the existence of multiple edges between nodes. For each graph, we construct multiple visualization variants by systematically varying the layout and semantic content of node and edge labels while preserving the underlying graph structure. This design enables controlled analysis of whether model performance depends on the graph structure itself or on the properties of its visualization, such as layout or semantic information contained in node and edge labels. Beyond task accuracy, VGH therefore supports fine-grained evaluation of visual graph perception, algorithmic reasoning, semantic sensitivity, robustness to visualization changes, and consistency across related graph tasks.

1 INTRODUCTION

Graphs provide a general representation for relational data, encoding entities as nodes and their relationships as edges. They arise across a wide range of domains, including social networks, knowledge bases, molecular structures, route maps, and mathematical diagrams. In many of these settings, graph-structured information is commonly encountered in visual form rather than as an explicit textual serialization. Route maps encode connectivity through spatial arrangements, molecular diagrams depict atoms and bonds graphically, knowledge graphs are often rendered as node-link diagrams, and mathematical objects such as commutative diagrams from topology use arrows, symbols, and layout to express relational structure. These visualizations make graph structure directly inspectable, but they also introduce a layer of visual perception that is absent when the same graph is provided as an adjacency list, edge list, or some other equivalent textual representation.

As vision-language models (VLMs) become increasingly capable of interpreting and reasoning over complex visual inputs, an important question is whether they can similarly understand and reason over graph-structured information presented as images. This capability is distinct from conventional image understanding. A model cannot directly apply graph reasoning to an image without first recovering the relevant graph structure: it must identify nodes and their labels, recognize edges and optionally edge labels when present, interpret their spatial relationships, and when applicable, distinguish different edge types or directions. Errors at this perceptual stage can propagate to downstream graph computations even when the underlying reasoning procedure is otherwise simple. Visual

[1] VGH is publicly available through an anonymized drive

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

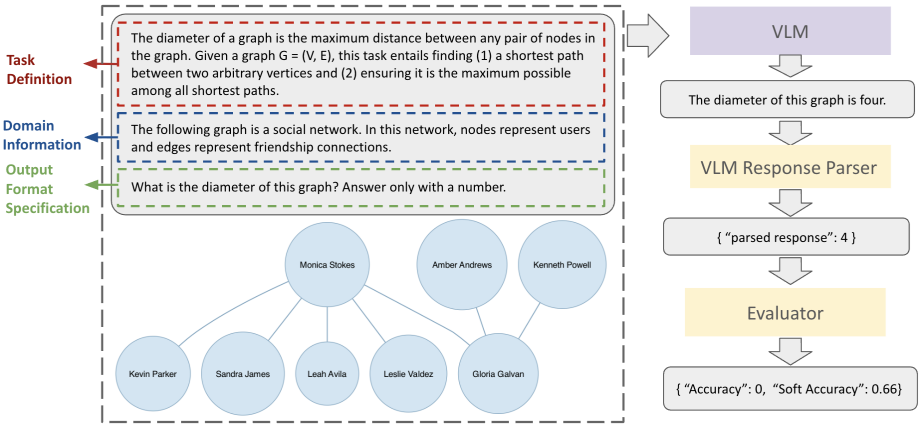


Figure 1: The workflow of evaluating a sample graph task and image in VGH. The prompt belongs to the task of finding the value of the diameter in the graph. The detailed prompts for the different tasks are discussed in Appendix E.

graph understanding therefore requires the joint evaluation of two capabilities: accurately extracting graph structure from visual input, and correctly reasoning over the recovered structure.

Existing graph reasoning benchmarks have largely studied the latter capability by representing graphs textually and evaluating large language models on graph-computational tasks. While such benchmarks are valuable for measuring reasoning over an explicitly specified graph structure, they remove the perceptual problem entirely. Evaluating only textual graph representations therefore provides an incomplete view of the challenges involved when graphs are presented visually. Recent work has begun to extend graph evaluation to VLMs by introducing image-based graph reasoning benchmarks (Babaiee et al., 2025; Li et al., 2024). However, existing benchmarks typically cover only a subset of the dimensions required for a comprehensive evaluation. As summarized in Table 2, they differ substantially in their coverage of read world graph domains, graph sizes, visualization styles, and reasoning tasks. Some focus primarily on synthetically generated graphs, while others consider a narrower set of graph problems or visual forms. Consequently, it remains difficult to determine whether a model that performs well in one setting has learned a general ability to interpret graph structure or has instead adapted to particular graph distributions, layouts, or task formats.

Beyond coverage, existing evaluations also provide limited mechanisms for diagnosing *why* a model succeeds or fails. Standard task accuracy does not distinguish failures of visual perception from failures of graph reasoning, nor does it reveal whether a model is exploiting semantic information contained in node and edge labels instead of reasoning over graph topology. Similarly, evaluating a single rendering of each graph makes it difficult to determine whether predictions are robust to changes in layout and visualization style. Different graph tasks can also impose logically related constraints on model outputs; for example, the number of nodes predicted by a node-counting task should agree with the number of nodes recovered by a node-listing task. Evaluating these predictions independently can obscure inconsistencies in the model's representation of the same underlying graph. A more informative benchmark should therefore complement task-level correctness with evaluation of partial correctness, semantic dependence, robustness to visualization changes, and consistency across related predictions.

In this work, we introduce Visual Graph Harness (VGH), a comprehensive benchmark for evaluating graph understanding and reasoning in VLMs. VGH combines real-world graphs spanning multiple domains, including knowledge graphs, route maps, social networks, and commutative diagrams from topology. The inclusion of commutative diagrams further introduces visual challenges such as mathematical notation, repeated node labels, and multiple edges. The benchmark covers a broad range of graph capabilities, beginning with fine-grained visual understanding tasks that assess whether models can recover graph structure from an image. These include identifying and counting nodes and edges, determining node degree, and recovering node and edge sets. We then evaluate algorithmic reasoning over the perceived graph through polynomial-time tasks such as shortest-path distance, graph diameter, and connected components, as well as computationally challenging

problems, including maximum clique, maximum independent set, minimum vertex cover, traveling salesman, and chordless cycle.

A central design goal of VGH is to enable controlled analysis rather than only aggregate benchmark comparison. We therefore construct multiple visualizations of the same underlying graph that vary in properties such as graph layout and semantic information while preserving graph structure. Comparing model behavior across these variants allows us to study robustness to changes in graph rendering and to distinguish structural reasoning from reliance on semantic priors encoded in node and edge labels. We additionally analyze performance as graph size changes, providing a means to study how visual graph understanding and reasoning scale with graph complexity.

Finally, VGH uses a multi-stage evaluation protocol designed to characterize model behavior beyond exact task accuracy. We first measure whether model responses can be parsed according to the requested output format. For valid responses, we report strict accuracy together with task-specific approximate accuracy, which provides a more fine-grained measure of correctness when predictions are close to, but do not exactly match, the ground truth. We further evaluate semantic sensitivity by comparing predictions on semantically labeled and anonymized versions of the same graph, and self-consistency by comparing outputs from logically related tasks. Together, these components enable a systematic evaluation of whether VLMs can *perceive*, *recover*, and *reason* over graph structures across graph domains, visualization styles, semantic conditions, and levels of computational complexity. We use VGH to evaluate a diverse collection of open-source VLMs and to analyze their behavior with respect to graph semantics, visualization, and graph size.

2 Related Work

LLMs on Graph Problems. Early work on evaluating graph reasoning by LLMs started with synthetic datasets. GraphWiz (Chen et al., 2024) was among the first to evaluate LLMs on graph reasoning tasks, covering both NP-complete and polynomial-time problems on synthetic graphs. NLGraph (Wang et al., 2023) further expanded this direction by introducing a synthetic benchmark for graph reasoning. These benchmarks are largely limited to synthetic graphs and their textual representations. Moreover, they lack the semantic richness and structural diversity of real world graph data. More recently, GraphArena (Tang et al., 2025) extended evaluation to real-world graphs.

Vision Language Models on Graphs. Several benchmarks have investigated graph reasoning using visual graph representations. VisionGraph (Li et al., 2024) visualizes the synthetic graphs from NLGraph using manually selected layouts and simple textual annotations (e.g., having numeric labels for nodes). GLVQA (Wei et al., 2024) created multiple visualizations per graph for the same underlying dataset. VisualGraphArena (Babaiee et al., 2025) evaluates VLMs on four NP-complete problems with a focus on having multiple layouts per graph, but is limited to synthetic graphs with only 8-9 nodes. SnorkelGraph (Snorkel AI, 2025), although not publicly available, follows a similar paradigm by visualizing graph problems and reports a collection of approximately 200 graph instances with varying complexity. In contrast, Graph-to-Vision (Li & Jiang, 2025) considers reasoning over pairs of graphs, such as graph isomorphism, using graphs in their original visual form (e.g., maps). Overall, existing benchmarks focus either on synthetic graph visualizations or specific reasoning tasks, leaving semantically grounded, real-world graph data largely underexplored.

Benchmarking Visual Reasoning in VLMs. Benchmarking the visual reasoning capabilities of VLMs has recently attracted considerable attention. PAM (Weng et al., 2026) evaluates visual cognition along the dimensions of perception, attention, and memory. Berman & Deng (2026) studies non-local spatial reasoning through tasks such as visual search and comparative perception. MME (Fu et al., 2026a) provides a comprehensive evaluation for perception and reasoning abilities of VLMs. More recent benchmarks focus on mathematical reasoning from images (Xu et al., 2026; Wang et al., 2026) and understanding text embedded in images (Fu et al., 2026a;b).

VLM Benchmarking. In the past few years, benchmarking the visual cognition capabilities of VLMs has gained traction. The PAM dataset (Weng et al., 2026) was introduced to evaluate VLMs along the cognitive axes of perception, attention, and memory. In Berman & Deng (2026), the authors evaluated the non-local spatial reasoning of VLMs, including tasks such as visual search and comparative perception. Fu et al. (2026a) included both perception and reasoning tasks in their

Table 1: Comparison of previous graph benchmark attributes against VGH. **Real Domains:** GraphArena → {DBPedia, OpenFlights, DBLP, PubChem}, Graph to Vision → {mindMap, knowledge graph, route map, flowchart}, VGH → {DBPedia, OpenFlights, DBLP, Toplogy}

Benchmark	Modality	Graph Size ($\|V\|$)	OCR	Fine-grained Metrics	Domain
NLGraph (Wang et al., 2023)	Text	≤ 35	-	No	Synthetic
GraphWiz (Chen et al., 2024)	Text	≤ 100	-	No	Synthetic
GraphArena (Tang et al., 2025)	Text	≤ 50	-	Yes	Real
GVLQA (Wei et al., 2024)	Text + Image	≤ 35	Yes	No	synthetic
VisualGraphArena (Babaiee et al., 2025)	Text + Image	8, 9	No	No	Synthetic
VisionGraph (Li et al., 2024)	Text + Image	≤ 35	Yes	No	Synthetic
Graph to Vision (Li & Jiang, 2025)	Text + Image	-	yes	yes	Real
SnorkelGraph (Snorkel AI, 2025)	Text + Image	-	Yes	No	Synthetic
Visual Graph Harness (VGH)	Text + Image	≤ 50	Yes	Yes	Real

Table 2: Summary of the graph understanding and reasoning tasks in VGH. Task description is defined over an image representing a graph $G = (V, E)$ (see Appendix A for formal definitions).

Task Type	**Task**	**Definition**
Graph Understanding	NODECHECK	Determine whether a queried node v belongs to V.
	NODEDEGREE	Return the degree of a queried node v.
	NODECOUNT	Return the number of nodes $\|V\|$.
	EDGECOUNT	Return the number of edges $\|E\|$.
Graph Reasoning (Polynomial time)	HIGHESTDEGREE	Return the degree of a highest degree node in G.
	SHORTESTDISTANCE	Return the shortest-path distance between two queried nodes u and v.
	DIAMETER	Return the diameter of G.
	CONNECTEDCOMPONENTCOUNT	Return number of connected components of G.
Graph Reasoning (NP-complete)	MAXIMUMCLIQUE	Return a maximum clique of G.
	MAXIMUMINDEPENDENTSET	Return a maximum independent set of G.
	TRAVELINGSALESMAN	Return a shortest Hamiltonian tour of G.
	MINIMUMVERTEXCOVER	Return a minimum vertex cover of G.
	CHORDLESSCYCLE	Return a chordless cycle of G.

comprehensive benchmark. More recently, this line of work has been extended to include rigorous mathematical reasoning from images (Xu et al., 2026; Wang et al., 2026), while another line of research has focused on understanding text embedded in images (Fu et al., 2026a;b). In contrast to these benchmarks, our work evaluates visual reasoning over graph-structured data, requiring models to jointly interpret visual layouts, graph structure, and semantic information.

3 VISUAL GRAPH HARNESS (VGH)

We introduce VGH, a benchmark designed to evaluate the graph understanding and reasoning capabilities of VLMs. VGH covers a wide range of real world graph types, diverse visualization configurations, and graph understanding tasks ranging from simple perception to complex reasoning, with both binary and descriptive responses. Beyond task diversity, VGH supports fine-grained evaluation through multiple metrics tailored to each task, enabling a more comprehensive assessment of model capabilities. Furthermore, the benchmark is designed to facilitate controlled analyses of model behavior by varying specific factors such as graph visualization layout, and semantic content. For example, we include anonymized versions of each graph, in which labels are replaced with generic placeholders, to evaluate models' reliance on semantic priors and memorized knowledge.

3.1 GRAPH TASKS

We evaluate graph visual understanding and reasoning. Unlike textual graph benchmarks (Tang et al., 2025; Chen et al., 2024), VGH represents graphs as images, requiring models to overcome visual perception challenges such as reading text embedded in the graph (e.g., node labels and edge labels), understanding the spatial relationships between nodes and edges, and interpreting diverse visualization styles. Since there is a modality gap between graph images and graph algorithms,

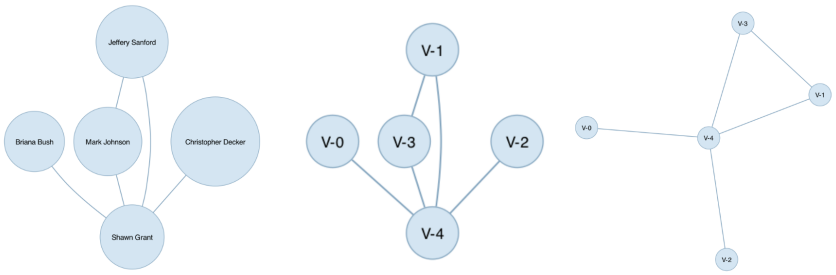


Figure 2: Sample visualization variations of a graph in VGH. The left picture is the original image of the graph with the DOT layout. The middle picture is a visualization of a simplified graph with the DOT layout, and the right one is the visualization of a simplified graph with the NEATO layout.

which we have observed to cause performance differences in current VLMs (Babaiee et al., 2025), our benchmark includes both understanding and reasoning tasks.

Graph Visual Understanding. To evaluate a model's ability to understand graph images, we include a set of graph visual understanding tasks that assess its ability to perceive the graph structure encoded in an image before performing any graph reasoning. These tasks focus on graph perception rather than algorithmic reasoning. They include NODECHECK, which determines whether a queried node exists in the graph; NODEDEGREE, which returns the degree of a queried node; NODECOUNT and EDGECOUNT, which return the number of nodes and edges, respectively.

Graph Reasoning. Beyond graph understanding, we include graph reasoning tasks that require algorithmic computations over the graph. These tasks vary in both response format (e.g., numerical answers, node lists, etc) and in computational complexity, allowing us to evaluate reasoning capabilities on tasks differentiated by their hardness. We categorize task difficulty according to the complexity of the underlying graph algorithm to solve the task: (a) tasks that require polynomial time computation, and (b) tasks that correspond to NP-complete problems. The polynomial time reasoning tasks include HIGHESTDEGREE, which identifies a node with maximum degree; SHORTESTDISTANCE, which computes the distance of the shortest path between two queried nodes; DIAMETER, which returns the graph diameter; and CONNECTEDCOMPONENTCOUNT, which returns the number of connected components in the graph. The NP-complete reasoning tasks include: MAXIMUMCLIQUE, which requires finding the largest clique in the graph; MAXIMUMINDEPENDENTSET, which requires finding the largest set of pairwise non-adjacent nodes; TRAVELINGSALESMAN, which requires finding the shortest Hamiltonian tour; MINIMUMVERTEXCOVER, which requires finding the smallest set of nodes that covers all edges; and CHORDLESSCYCLE, which requires identifying a cycle with no chords. Together, these tasks evaluate a model's ability to reason over computationally challenging graph problems beyond basic graph analysis. Formal definitions of all tasks are provided in Table 3 and Appendix A. For each task, we adopt prompts from prior work whenever available, making only minimal modifications. The structure of prompts is as follows: a description of the problem to be solved (not applicable for understanding tasks), a brief description of the graph domain (when applicable), followed by the task instruction and output format. The input to each model consists of the prompt, together with the graph image. Figure 1 illustrates a sample prompt and image in VGH (see Appendix E for more details on prompts).

3.2 GRAPH DATA

The graphs included in VGH are collected from prior work on evaluating LLMs on graph tasks. Specifically, we collect graphs from real-world datasets, e.g. GraphArena (Tang et al., 2025) and topology textbooks in mathematics (Hatcher, 2002; Rotman & Rotman, 2009; Switzer et al., 1975). We include graphs from 3 domains from GraphArena: knowledge graphs, route maps, and social networks. We further introduce a new domain consisting of *commutative diagrams* from topology. Commutative diagrams are an important mathematical concept that can be represented as graphs; see Appendix B for details on the commutative diagrams part of our dataset. Compared to the other domains, these graphs introduce additional challenges, including mathematical symbols as node or edge labels, multiple edge types, and repeated node names. Though commutative diagrams can be

Table 3: Performance of VLMs on the graphs with images created by DOT visualization layout. For each task category, the macro-average value is reported across the tasks in that category.

Model	Graph Understanding			Graph Reasoning (Polynomial Time)			Graph Reasoning (NP-Complete)		
	Parsable	Accuracy	Soft Accuracy	Parsable	Accuracy	Soft Accuracy	Parsable	Accuracy	Soft Accuracy
Qwen3-VL-32B	97.9	61.8	82.5	96.2	49.7	70.0	94.7	12.7	28.4
Qwen3-VL-8B	98.5	61.4	82.7	97.1	54.8	72.5	89.2	7.5	24.2
InternVL-30B	97.0	54.4	76.3	98.7	54.3	72.9	95.7	7.5	17.0
InternVL-14B	90.3	48.6	76.4	99.2	49.1	69.5	90.9	6.7	23.5
InternVL-8B	98.7	57.7	83.2	99.0	47.7	67.8	98.7	5.9	25.2
Gemma3-27B	98.8	51.1	77.7	98.7	41.3	60.0	98.6	6.2	24.7
Gemma3-8B	100.0	46.6	76.4	100.0	23.3	44.6	100.0	4.2	36.2
Gemma4-8B	99.7	47.0	75.6	100.0	43.1	66.2	98.0	8.0	26.1
LLaVA-13B	91.9	37.7	67.1	84.1	14.9	34.3	82.2	3.3	20.5
LLaVA-7B	99.8	34.1	59.6	97.2	20.6	39.9	99.9	3.6	19.0
BakLLaVA	99.5	20.0	49.2	98.8	16.3	40.4	100.0	4.4	20.2
GLM	99.1	60.1	82.9	100.0	48.0	68.0	100.0	7.1	24.3
Molmo2	99.8	56.2	79.8	100.0	50.1	71.2	98.7	5.6	23.5
DeepSeekTiny	99.9	45.6	72.0	99.7	28.1	46.9	97.0	7.5	20.8
MiniCPM	98.6	56.9	80.7	99.9	42.1	60.4	98.1	6.9	25.1

naturally represented as graphs, they are typically presented as mathematical diagrams rather than explicitly modeled as graph data. Therefore, we manually extracted commutative diagrams from the topology textbooks mentioned above and represent their underlying structures as graphs.

To avoid duplication, we remove isomorphic graphs from the final benchmark, resulting in 8251 graphs. The number of nodes in these graphs ranges from 3 to 50. Among them, 1000 are weighted, and 1374 have non-numeric edge labels. For each graph, we provide three visualization variants using GraphViz (Ellson et al., 2004): (1) an image created with the DOT (Gansner, 2006) layout, (2) an image created with the DOT layout and anonymized node and edge labels, and (3) an image created with the NEATO (Gansner et al., 2005) layout, anoomized node and edge labels. In the anonymized version of the graphs, we replace node labels and edge labels with generic labels, i.e., node i is labeled with V-i and edge i is labeled with e-i (if the edge i has a label in the graph). These variants enable controlled analyses of model robustness to changes in visualization and semantic content while preserving the underlying graph structure. Figure 2 illustrates the three visualization variants for a sample graph.

3.3 EVALUATION

Each model receives the task prompt together with the graph image as input. To ensure fair evaluation and avoid unbounded generation, we assign task-specific generation token budget proportional to the expected output size. Graph understanding tasks receive smaller generation budgets (i.e., 256 tokens), whereas reasoning tasks that require generating node or edge sequences are allocated larger budgets. In particular, polynomial-time graph reasoning tasks receive a token budget of 256, and NP-complete reasoning tasks have a maximum token limit of 2048. Responses exceeding the generation budget or passed time limit are terminated and marked as missing. All evaluations are performed using greedy, deterministic decoding by setting the temperature parameter to zero.

We use task-specific generation budgets to control inference-time computation and ensure fair comparison across models. Prior work shows that token usage increases with problem difficulty Yang et al. (2025) and that model performance can vary substantially with the allowed generation budget de Souza & Panisson (2026). We therefore use 256 tokens for graph understanding and polynomial-time tasks, and 2048 tokens for NP-complete tasks, which may require longer reasoning and structured outputs.

We evaluate model outputs using a multi-stage evaluation protocol. We first report the percentage of *parsable* responses, i.e., responses that can be successfully extracted using pattern matching according to the expected output format. Responses that cannot be parsed, either because they do not comply with the required format or exceed the generation budget or time limit, are considered invalid and are excluded from subsequent task-specific evaluation. For example, for the task of MAXIMUMCLIQUE with an expected output format of node list, the answer '[V-1 to V-5]' is not parsable, whereas the answer "The maximum clique is [V-1, V-2, V-3, V-4, V-5]" is parsable.

For parsable responses, we perform task-specific verification. Each task is evaluated using two complementary metrics: *strict accuracy* and *soft (approximate) accuracy*. Strict accuracy measures

whether the predicted answer exactly matches the ground-truth answer, whereas soft accuracy is task-dependent and measures how close the predicted answer is to the correct answer. For example, for the DIAMETER task, we compute the relative distance between the predicted graph diameter and the ground-truth value. Soft accuracy provides a more fine-grained assessment of model capabilities, particularly for tasks where partially correct answers remain informative. It also provides greater distinguishability on more challenging tasks, where exact matches are rare. Formally, for an instance x with ground truth y, if the predicted answer is $\hat{y}$, we define approximate accuracy as $1 - \min\left(\frac{|y-\hat{y}|}{|y|}, 1\right)$.

In addition to task-specific correctness, we measure *semantic sensitivity* of models by comparing model performance on the original and anonymized (node labels and edge labels changed to simple letters) versions of the same graph, thereby quantifying the extent to which models rely on semantic priors rather than graph structure.

4 EXPERIMENTS

We follow previous benchmarking efforts and evaluate a diverse set of open-source VLMs using VGH, spanning multiple model families and parameter size (1 to 32 billion). The list of models and details about them is available in Table 4. We report overall benchmark performance, then we use the controlled design of VGH to analyze the effects of different aspects of the dataset: graph semantics, visualization, and graph size. These experiments demonstrate that VGH enables fine-grained analyses of model behavior beyond standard benchmark leaderboards and provides insights into the strengths and limitations of current open source VLMs.

4.1 VLMS PERFORMANCE ON GRAPH TASKS

We evaluate all models on the split of data with the DOT layout visualization of the graph. For each task, we report the percentage of parsable responses (responses that follow the instruction in output format), strict accuracy, and approximate accuracy. We report the macro-average performance within each task category in Table 6. Refer to the Appendix C for the fine-grained evaluation of the models for each task. We observe that most models yield a high parsable rate (higher than 95 percent) on all tasks, even where their task accuracy is extremely low. Therefore, the lower performance on graph reasoning tasks cannot be attributed to a failure to follow the output specification; it reflects a failure in the underlying task. As shown in Table 6, VLMs performance consistently decreases as the tasks become more complex. Across nearly all models, performance follows a pattern: graph understanding achieves the highest accuracy, followed by polynomial-time graph reasoning, while performance drops substantially on NP-complete reasoning tasks. In particular, exact accuracy on NP-complete tasks remains below 13% for every evaluated model, despite several models exceeding 60% accuracy on graph understanding. This suggests that successfully extracting and understanding graph structure from an image does not necessarily translate into the ability to perform complex reasoning over that structure.

Surprisingly, increasing model size within the same model family does not consistently show better performance. Despite the general expectation that larger models should provide stronger reasoning (Nezhurina et al., 2025), we observe only small gains from scaling within a model family. In several cases, smaller variants even outperform their larger counterparts; for example, achieves higher polynomial-time reasoning accuracy than (54.8 vs. 49.7). Similar non-monotonic patterns appear across the InternVL and Gemma families. This suggests that visual graph reasoning may not benefit from model scaling in the same way as more general vision-language capabilities.

We observe a non-negligible gap between exact accuracy and soft accuracy, especially for more difficult tasks. While models often fail to produce the fully correct answer, their much higher soft accuracy shows that they can still solve parts of the problem correctly. This gap is particularly clear for NP-complete tasks, where exact accuracy is very low across all models, but soft accuracy remains noticeably higher. For example, Gemma3-8B achieves only 4.2% exact accuracy on NP-complete tasks but reaches 36.2% soft accuracy. More generally, while exact NP-complete accuracy is uniformly very low, soft accuracy remains considerably higher (roughly 17–36%). This indicates that many failures are not completely incorrect solutions. This shows the importance of fine-grained evaluation, as exact accuracy alone does not fully capture the models' ability to reason over graphs.

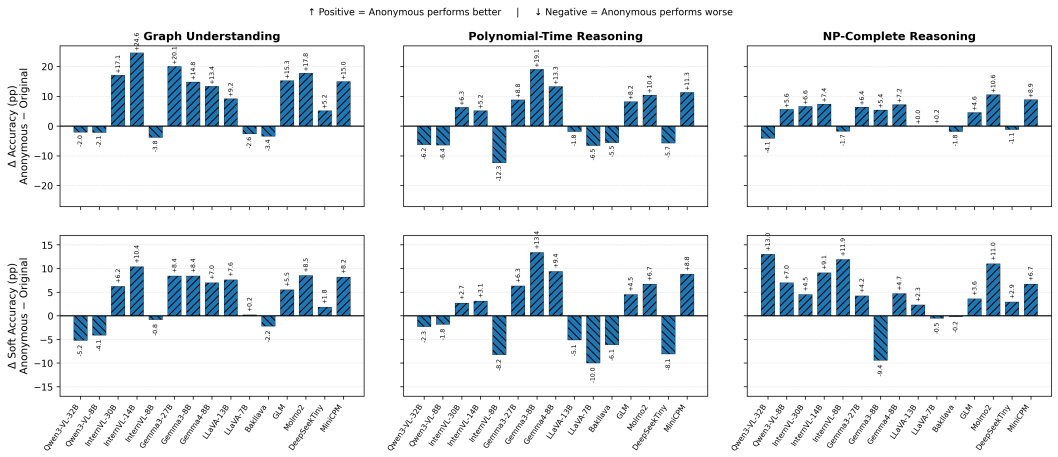


Figure 3: Performance gain under removing graph semantic information. The bars indicate the difference in model performance between anonymized graphs and the original visualization of graphs.

4.2 Effect of Graph Semantics

In this experiment, we investigate the VLMs reliance on semantic information encoded in node and edge labels, rather than the underlying graph structure. We evaluate all models on pairs of graphs with identical underlying graph structure and visualization layouts, where one version contains the original node and edge labels while the other contains anonymized labels. By comparing performance across these two settings, we quantify the semantic sensitivity of each model and assess whether graph reasoning is driven by structural understanding or by semantic priors.

The results are shown in Figure 3 and Table 6 in the Appendix. We observe that anonymization generally has a stronger positive effect on graph understanding tasks, with most models achieving higher accuracy on graphs with anonymous labels. This is possibly because anonymous labels are visually simpler and less semantically distracting, reducing the perceptual burden of identifying and referring to nodes. However, this trend is not universal: the Qwen models, for example, show a small degradation after anonymization. Moreover, models differ substantially in their sensitivity to anonymization. Some models, such as InternVL-14B, InternVL-30B, Gemma family models, Molmo2, and MiniCPM often benefit considerably from anonymous labels, whereas others show smaller or less consistent changes. This suggests that model performance can depend not only on the underlying graph structure, but also on how node semantics are visually represented. We also observe that the effect of anonymization depends on task complexity. While the improvement is particularly prominent for graph understanding, the effect becomes more heterogeneous for polynomial-time reasoning, where anonymization improves some models but degrades others. For NP-complete reasoning, exact accuracy remains low under both settings, although anonymous graphs often yield noticeable improvements in soft accuracy. This distinction suggests that anonymization may help models recover more partially correct reasoning even when it is insufficient to produce a fully correct solution to difficult graph problems.

We investigate whether semantic sensitivity is consistent across graph domains. While anonymization removes semantic priors in every domain, its effect may vary depending on the extent to which node and edge labels encode meaningful domain knowledge. Refer to the Appendix C.1 for details of this analysis.

4.3 Effect of Graph Visualization

In this experiment, we investigate the sensitivity of VLMs to different graph visualizations. We evaluate each model on multiple visual representations of the same underlying graph while keeping the graph structure and task unchanged. Specifically, we compare two visualization styles: the Graphviz DOT layout against the Graphviz NEATO layout. By isolating the visualization from the underlying graph, this experiment evaluates the robustness of VLMs to changes in graph layout and rendering. The results are summarized in Figure 4.

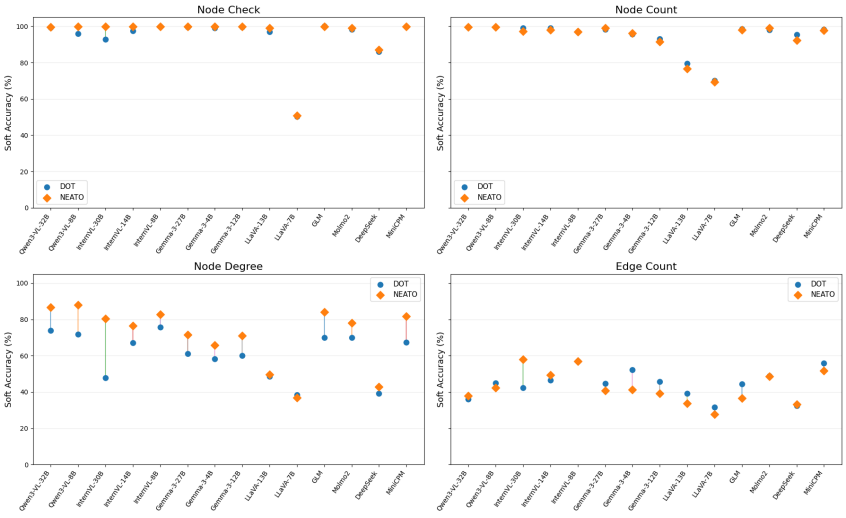


Figure 4: Effect of graph-visualization layout on the performance of VLMs on visual-understanding tasks with anonymized node labels. The plots show the value of soft accuracy for each task.

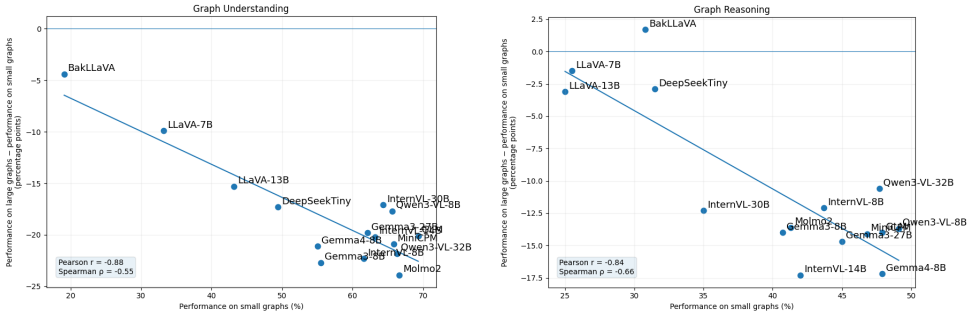


Figure 5: Effect of graph size on VLM performance. The plots show the correlation of the model performance gap on small graphs and large graphs, and its correlation with model performance.

4.4 Effect of graph size

In this experiment, we investigate how graph size affects the graph understanding and reasoning capabilities of VLMs. Similar to Tang et al. (2025), we group graphs according to the number of nodes: easy graphs (with less than 20 nodes) and hard graphs (with more than 20 nodes), and evaluate model performance within each size category. This analysis reveals how well current VLMs scale as graph complexity increases and whether larger graphs pose greater challenges for visual perception, structural understanding, and graph reasoning. The results are shown in Figure 5. We observe that a large graph size consistently negatively impacts all models' performance. The impact is more prominent for graph understanding tasks, where most models show at least a 17% performance drop. Moreover, models that perform better on small graphs generally experience larger absolute performance drops as graph size increases.

5 Conclusions

In this work, we introduced VGH, a benchmark for evaluating visual graph understanding and reasoning in VLMs across real-world graph domains, diverse visualizations, and tasks spanning perception to computationally challenging graph reasoning. Our experiments show that performance decreases substantially as task complexity increases, with particularly low exact accuracy on NP-complete problems, while higher soft accuracy reveals that models often recover partially correct solutions. We further find that performance is sensitive to graph semantics and graph size, and that increasing model scale does not consistently improve visual graph reasoning. Overall, these results highlight a substantial gap between perceiving graph structure and reliably reasoning over it, and establish VGH as a framework for systematically studying this gap.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript, search for related work, and refactor and review the benchmark's codebase. The code was reviewed and tested by two authors for clarity and correctness. Moreover, we used generative AI tools to assist with the quality assessment of the generated images, manually reviewed unclear images, and made adjustments. We also leveraged generative AI tools to help identify unexpected error patterns in the model responses during the experiment.

REPRODUCIBILITY STATEMENT

We publicly release the code and benchmark dataset used in this work. We provide detailed information and code about the benchmark construction, evaluation protocols, and experimental settings. For all evaluated models, we report their model identifiers (IDs). Moreover, relevant experimental configurations are highlighted in the code, enabling researchers to reproduce our results using the same model versions and settings. Moreover, we have included instructions for researchers to evaluate their models. The released code also includes the scripts necessary to run the benchmark and reproduce the reported evaluation results.

REFERENCES

Zahra Babaiee, Peyman Kiasari, Daniela Rus, and Radu Grosu. Visual Graph Arena: Evaluating Visual Conceptualization of Vision and Multimodal Large Language Models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=BCJPAmlfxv.

Shmuel Berman and Jia Deng. VLMs have Tunnel Vision: Evaluating Nonlocal Visual Reasoning in Leading VLMs. *Advances in Neural Information Processing Systems*, 38:78972–78993, 2026.

Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 353–364, 2024.

Rodrigo Guedes de Souza and Alison R Panisson. Who Thinks Best Depends on How Long You Let Them: Budget-Dependent Rankings in LLM Evaluation. *arXiv preprint arXiv:2608.12150*, 2026.

John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. *Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools*, pp. 127–148. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-642-18638-7. doi: 10.1007/978-3-642-18638-7_6. URL https://doi.org/10.1007/978-3-642-18638-7_6.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *Advances in Neural Information Processing Systems*, 38, 2026a.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning. *Advances in Neural Information Processing Systems*, 38, 2026b.

Emden R. Gansner. Drawing graphs with dot. 2006. URL https://api.semanticscholar.org/CorpusID:60720985.

Emden R. Gansner, Yehuda Koren, and Stephen North. Graph drawing by stress majorization. In János Pach (ed.), *Graph Drawing*, pp. 239–250, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31843-9.

Allen Hatcher. *Algebraic topology*, volume 1. Cambridge university press Cambridge, 2002.

Ruizhou Li and Haiyun Jiang. Graph-to-Vision: Multi-graph Understanding and Reasoning using Vision-Language Models, 2025. URL https://arxiv.org/abs/2503.21435.

Yunxin Li, Baotian Hu, Haoyuan Shi, Wei Wang, Longyue Wang, and Min Zhang. VisionGraph: Leveraging Large Multimodal Models for Graph Theory Problems in Visual Context. *arXiv preprint arXiv:2405.04950*, 2024.

Marianna Nezhurina, Tomer Porian, Giovanni Puccetti, Tommie Kerssies, Romain Beaumont, Mehdi Cherti, and Jenia Jitsev. Scaling laws for robust comparison of open foundation language-vision models and datasets. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 153979–154018. Curran Associates, Inc., 2025. doi: 10.52202/085713-5148. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e2470f02531c063776ad240306b5cfef-Paper-Conference.pdf.

Joseph J Rotman and Joseph J Rotman. *An introduction to homological algebra*, volume 2. Springer, 2009.

Snorkel AI. SnorkelGraph: Graph Reasoning Benchmark for LLMs, 2025. URL https://snorkel.ai/leaderboard/snorkel-graph/.

Robert M Switzer et al. *Algebraic topology: homotopy and homology*, volume 212. Springer, 1975.

Jianheng Tang, Qifan Zhang, Yuhan Li, Nuo Chen, and Jia Li. GraphArena: Evaluating and Improving Large Language Models on Graph Computation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Y1r9yCMzeA.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can Language Models Solve Graph Problems in Natural Language? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=UDqHhbqYJV.

Peijie Wang, Chao Yang, Zhong-Zhi Li, Dekang Ran, Mi Tian, Zhilong Ji, Jinfeng Bai, Chenglin Liu, et al. SolidGeo: Measuring Multimodal Spatial Math Reasoning in Solid Geometry. *Advances in Neural Information Processing Systems*, 38, 2026.

Yanbin Wei, Shuai Fu, Weisen Jiang, Zejian Zhang, Zhixiong Zeng, Qi Wu, James Kwok, and Yu Zhang. GITA: Graph to Visual and Textual Integration for Vision-Language Graph Reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Zihan Weng, Lucas Gomez, Taylor Webb, and Pouya Bashivan. Caption This, Reason That: VLMs Caught in the Middle. *Advances in Neural Information Processing Systems*, 38:39891–39930, 2026.

Rui Xu, Dakuan Lu, Zicheng Zhao, Xiaoyu Tan, Xintao Wang, Siyu Yuan, Jiangjie Chen, et al. ORIGAMISPACE: Benchmarking Multimodal LLMs in Multi-Step Spatial Reasoning with Mathematical Constraints. *Advances in Neural Information Processing Systems*, 38:143108–143142, 2026.

Chang Yang, Ruiyu Wang, Junzhe Jiang, Qi Jiang, Qinggang Zhang, Yanchen Deng, Shuxin Li, Shuyue Hu, Bo Li, Florian T Pokorny, Xiao Huang, and Xinrun Wang. Nondeterministic Polynomial-time Problem Challenge: An Ever-Scaling Reasoning Benchmark for LLMs. *arXiv preprint arXiv:2504.11239*, 2025.

A Graph Task Formal Definitions

Let $G = (V, E)$ denote the input graph presented as an image, where V is the set of nodes (or vertices) and $E \subseteq \{(u, v) \in V^2, u \neq v\}$ is the set of edges. In this paper, we assume that any graph G is always an undirected graph, unless specified otherwise, and thus we assume that $(u, v) \in E$ implies $(v, u) \in E$. For $v \in V$, let $N(v) := \{u \in V : (u, v) \in E\}$ denote the neighborhood of v, and let $\deg(v) := |N(v)|$ denote its degree. Associated to each node in the graph images in

VGH is a label called *node label*, while edges can have multiple labels associated with them called *edge labels*. In VGH, only the part of the dataset associated with commutative diagrams can have multiple edge labels. Sometimes, the node and edge labels can also be empty.

Graphs can be weighted or unweighted — when the graph is weighted, there is a weight function $w : E \to \mathbb{R}$, and the weights are positive. When the graph is unweighted, we define the weight function to be $w(e) = 1$, for all $e \in E$. In weighted graphs, the weights are indicated as edge labels in the images of the graph. In unweighted graphs, the edge labels are either absent or not numbers.

A path in the graph G is a sequence of distinct nodes $(v_0, v_1, \ldots, v_k)$ such that every pair of consecutive nodes is connected by an edge:

$$(v_{i-1}, v_i) \in E \quad \text{for all } i = 1, \ldots, k.$$

The nodes v_0 and v_k are called the endpoints of the path, and the length of the path is the number of edges it contains, i.e., $|P| = k$. Moreover, we write $P : v_0 \rightsquigarrow v_k$ to denote that a path P exists with v_0 and v_k as endpoints. A *cycle* is a path whose endpoints are the same.

For $u, v \in V$, we define a distance function

$$d_G(u, v) := \begin{cases} \min\limits_{P : u \rightsquigarrow v} \sum\limits_{e \in P} w(e) & \text{if } u \neq v, \;\; P : u \rightsquigarrow v \text{ exists}, \\ 0 & u = v, \\ \infty & \text{otherwise.} \end{cases}$$

A path $P : u \rightsquigarrow v$ that achieves $d_G(u, v) = \sum_{e \in P} w(e)$ is called a *shortest path*, while $d_G(u, v)$ is called the *shortest distance*, between u and v.

We now give the formal definitions of the benchmark tasks below:

(i) NODECHECK: Given a queried node label v, determine whether the corresponding node is present in the graph. Formally, the task is to compute $\mathbf{1}[v \in V]$, where $\mathbf{1}[\cdot]$ is the indicator function.

(ii) NODECOUNT: Determine the total number of nodes in the graph $n = |V|$.

(iii) NODEDEGREE: Given a queried node $v \in V$, determine the number of edges incident to v:

$$\deg(v) = |\{u \in V : (u, v) \in E\}| = |N(v)|.$$

(iv) EDGECOUNT: Determine the total number of edges in the graph $m = |E|$.

(v) HIGHESTDEGREE: Determine the maximum degree attained by any node in the graph:

$$\Delta(G) = \max_{v \in V} \deg(v).$$

(vi) CONNECTEDCOMPONENTCOUNT: We define an equivalence relation $\sim$ on G as follows: two vertices $u, v \in V$ are equivalent, i.e. $u \sim v$, if and only if there exists a path between u and v. Then the equivalence classes $V/\sim$ are called the *connected components* of G. The task is to determine the number of connected components of G.

(vii) DIAMETER: Determine the *diameter* of G, which is the maximum of the shortest distance between all pairs of nodes $u, v \in V$:

$$\operatorname{diam}(G) = \max_{u, v \in V} d_G(u, v).$$

For a disconnected graph, the diameter is infinity.

(viii) SHORTESTDISTANCE: Given nodes $u, v \in V$, determine the shortest distance $d_G(u, v)$ between them.

(ix) MAXIMUMCLIQUE: A subset $C \subseteq V$ is a *clique* if every pair of distinct nodes in C is connected by an edge:

$$\forall u, v \in C, \quad u \neq v \implies (u, v) \in E \quad \text{or} \quad (v, u) \in E.$$

The task is to find a clique of maximum cardinality:

$$C^\star \in \arg \max_{C \text{ is a clique}} |C|.$$

(x) MINIMUMVERTEXCOVER: A subset $C \subseteq V$ is a *vertex cover* if every edge has at least one endpoint in C:

$$\forall (u, v) \in E, \quad u \in C \text{ or } v \in C.$$

The task is to find a vertex cover of minimum cardinality:

$$C^\star \in \arg \min_{C \text{ is a vertex cover}} |C|.$$

(xi) MAXIMUMINDEPENDENTSET: A subset $I \subseteq V$ is an *independent set* if no two distinct nodes in I are adjacent:

$$\forall u, v \in I, \quad u \neq v \implies (u, v) \notin E.$$

The task is to find an independent set of maximum cardinality:

$$I^\star \in \arg \max_{I \text{ is an independent set}} |I|.$$

(xii) TRAVELINGSALESMAN: Find a minimum-weight *Hamiltonian cycle*, i.e., a cycle that visits every node exactly once before returning to its starting node. Formally,

$$H^\star \in \arg \min_{H \text{ is a Hamiltonian cycle}} \sum_{e \in H} w(e).$$

For an unweighted graph, this task may instead be interpreted as determining whether a Hamiltonian cycle exists (or returning such a cycle), since all Hamiltonian cycles contain exactly $|V|$ edges and therefore have equal cost under unit edge weights.

(xiii) CHORDLESSCYCLE: A cycle $C = (v_1, \ldots, v_k, v_1)$, with $k \geq 4$, is *chordless* if no two non-consecutive nodes on the cycle are *adjacent*, i.e., connected by an edge. Equivalently, the subgraph induced by the nodes of C contains exactly the edges of the cycle. The task is to find a chordless cycle of maximum length:

$$C^\star \in \arg \max_{C \text{ is a chordless cycle}} |C|,$$

where $|C|$ denotes the number of distinct nodes in the cycle. If no chordless cycle exists, the task returns that none exists.

B COMMUTATIVE DIAGRAMS IN TOPOLOGY

Commutative diagrams are graphical representations used throughout mathematics to describe relationships among mathematical objects and mappings between them. They are particularly common in areas such as algebraic topology and homological algebra, which are the sources of the commutative diagrams included in VGH; in particular, the source of the dataset are three math textbooks (Hatcher, 2002; Rotman & Rotman, 2009; Switzer et al., 1975). In a commutative diagram, nodes represent mathematical objects, for example, sets, groups, vector spaces, rings, or topological spaces; while directed arrows represent functions or, more generally, morphisms between these objects.

The defining feature of a commutative diagram is that different directed paths connecting the same pair of objects represent equivalent compositions of mappings; see Figure 6 for an example. In fact, some of the diagrams we consider are not technically commutative diagrams, as there are not multiple paths between nodes. For example, the diagram

$$0 \longrightarrow A \xrightarrow{f} B \xrightarrow{g} C \longrightarrow 0.$$

could represent what is known as a *short exact sequence* in topology, or simply a sequence of maps between different spaces; but for simplicity, and since this distinction is not relevant for our work, we will group all such diagrams under the category of commutative diagrams. Thus, unlike a generic directed graph, the spatial organization and arrow annotations in a commutative diagram are intended to encode mathematical relationships between maps. For the purposes of VGH, however, we treat a commutative diagram primarily as a visually represented graph: mathematical objects correspond to nodes and mappings correspond to edges. This representation makes it possible to apply the same graph-understanding and graph-reasoning tasks used elsewhere in the benchmark while retaining the distinctive visual properties of mathematical diagrams.

Commutative diagrams introduce several challenges that are uncommon, or substantially less pronounced, in standard node-link graph visualizations. First, node labels can be mathematically complex. Rather than short names or identifiers, an object may be represented by an expression containing subscripts, superscripts, parentheses, operators, or other mathematical notation. Edge labels can be similarly complex, since arrows may be annotated by the names of functions, induced maps, compositions, or mathematical symbols. Consequently, correctly recovering the graph requires not only locating nodes and arrows but also accurately interpreting fine-grained mathematical typography.

Second, node labels need not uniquely identify nodes. The same mathematical expression may occur at multiple spatial locations in a single diagram. Two visually distinct nodes can therefore have identical textual labels even though they occupy different positions and participate in different edges. This differs from many conventional graph datasets, where a node name acts as a unique identifier. Repeated labels introduce ambiguity for tasks that refer to nodes only by their displayed names and can lead to multiple valid interpretations or ground-truth answers.

Third, commutative diagrams may contain multiple edges between the same pair of nodes and multiple visually distinct edge types. An edge is not necessarily represented by a single conventional solid arrow: mathematical diagrams can use different arrow styles and symbols to convey different relationships. Accurate graph recovery, therefore, requires distinguishing the existence and endpoints of edges even when their graphical realization differs from the standard arrow conventions encountered elsewhere in VGH.

We include commutative diagrams because they provide a qualitatively different test of visual graph understanding from the other VGH domains. Knowledge graphs, social networks, and route networks generally use visually explicit nodes and comparatively conventional edges. Commutative diagrams instead combine graph structure with dense mathematical notation, repeated labels, heterogeneous arrow representations, and potentially multiple edges between nodes. As a result, they place additional demands on OCR, symbol recognition, edge tracing, and the association of labels with graph elements.

At the same time, the underlying graph abstraction allows these diagrams to participate in VGH's common evaluation framework. Once the relevant nodes and edges have been recovered from an image, the resulting structure can be evaluated using the graph-understanding and graph-reasoning tasks defined in Appendix A. This separation is useful for the central question studied by VGH: whether a model can first recover graph structure from a challenging visual representation and subsequently reason over that structure. Commutative diagrams provide an especially demanding instance of this perception-to-reasoning pipeline because successful graph recovery requires interpreting both graphical connectivity and specialized mathematical notation. We refer the reader to Figure 6 for examples of commutative diagrams and their corresponding visualization in VGH.

C Experiment Details

C.1 Semantic Sensitivity by Graph Domain

To examine whether the effect of node-label anonymization varies across graph semantics, we report performance separately for five domains: social networks, academic networks, knowledge graphs, flight-route networks, and commutative diagrams. Within each domain, we compare original and anonymized node labels across graph-understanding, polynomial-time graph-reasoning, and NP-hard graph-reasoning tasks. Figure 8 illustrates the performance difference caused by anonymization of graphs in different domains.

Across domains, we observe the highest sensitivity to node-label anonymization on commutative diagrams, where the performance differences between original and anonymized graphs are substantially larger for many models. We hypothesize that this increased sensitivity stems from the greater visual and structural complexity of commutative diagrams, which may make models more susceptible to changes in node-label representation. In contrast, the effect of anonymization is generally smaller and more consistent across the other graph domains.

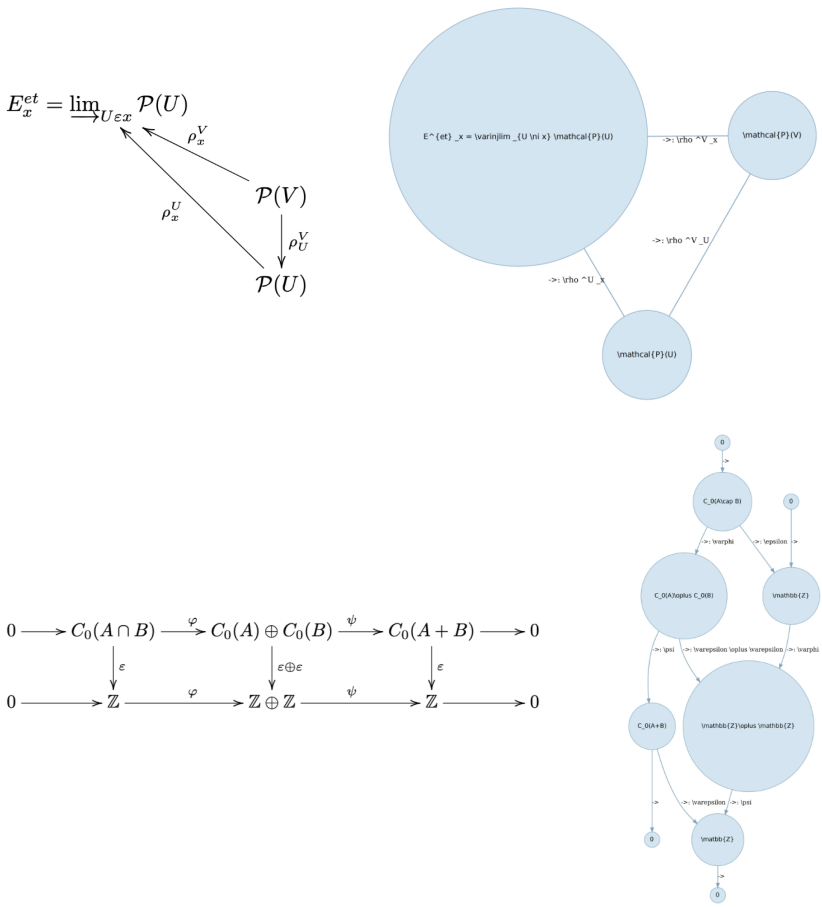


Figure 6: Sample commutative diagrams (on the left) and their visualization in VGH.

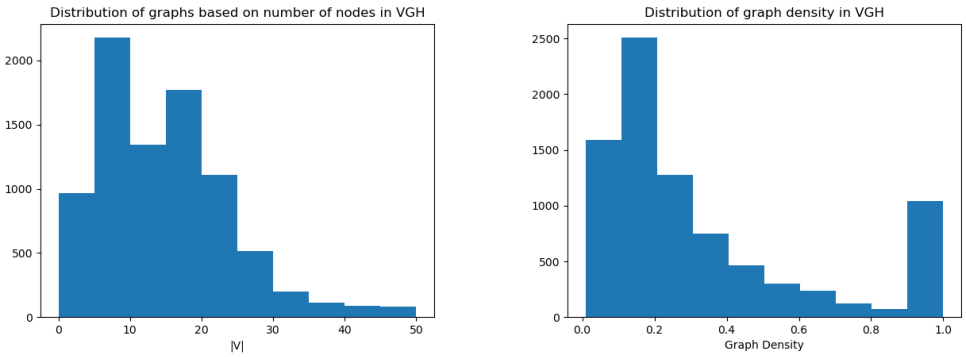


Figure 7: Statistics of graphs in VGH

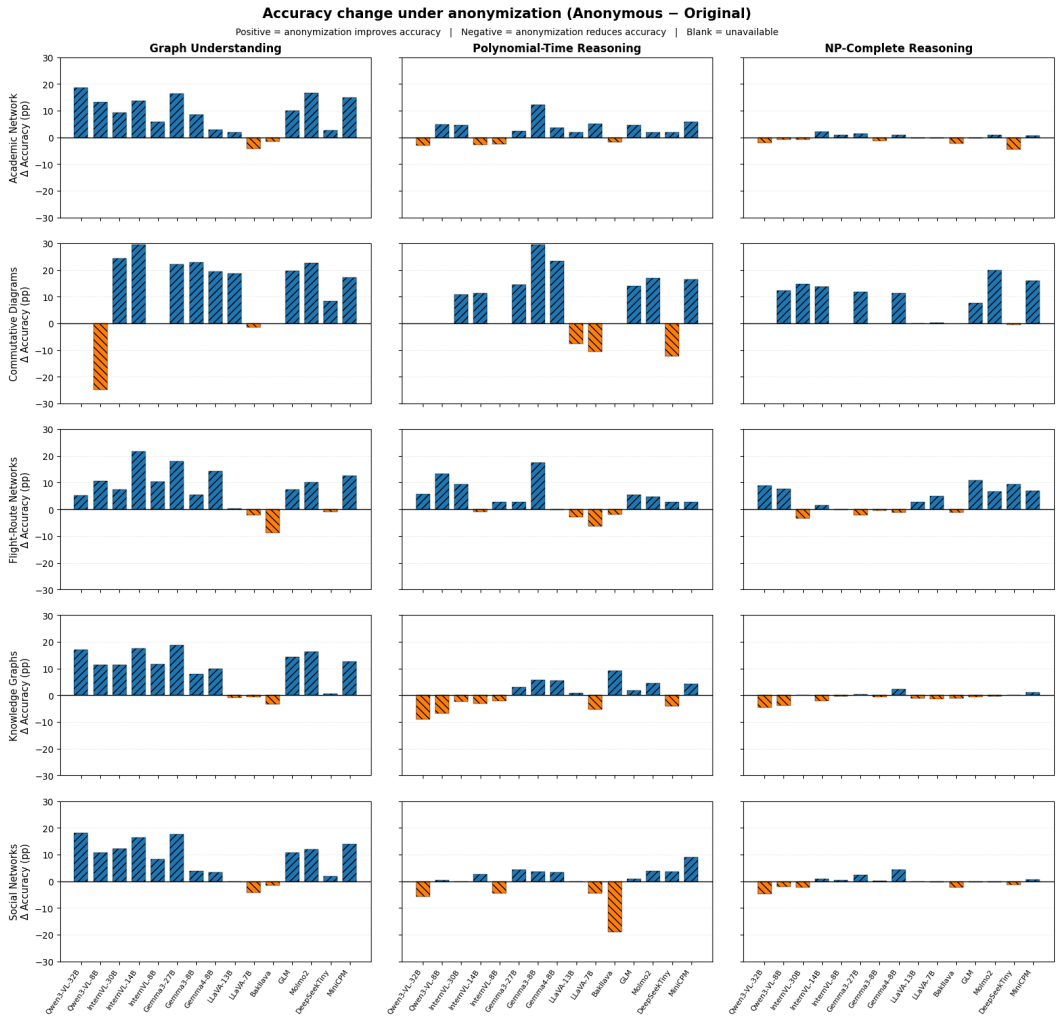


Figure 8: Performance gain under removing graph semantic information in different domains. . The bars indicate the difference in model performance between anonymized graphs and the original visualization of graphs.

Table 4: The list of VLMs in our experiments, along with their HuggingFace model IDs and parameter counts.

Model	# Parameters	Huggingface ID
Qwen3-VL-32B	33B	`Qwen/Qwen3-VL-32B-Instruct`
Qwen3-VL-8B	8B	`Qwen/Qwen3-VL-8B-Instruct`
InternVL-30B	31B	`OpenGVLab/InternVL3_5-30B-A3B-HF`
InternVL-14B	15B	`OpenGVLab/InternVL3_5-14B-HF`
InternVL-8B	9B	`OpenGVLab/InternVL3_5-8B-HF`
Gemma3-27B	27B	`google/gemma-3-27b-it`
Gemma3-8B	8B	`google/gemma-3n-E4B-it`
Gemma4-8B	8B	`google/gemma-4-E4B-it`
LLaVA-13B	13B	`llava-hf/llava-v1.6-vicuna-13b-hf`
LLaVA-7B	7B	`llava-hf/llava-v1.6-vicuna-7b-hf`
BakLLaVA	8B	`llava-hf/bakLlava-v1-hf`
GLM	10B	`zai-org/GLM-4.6V-Flash`
Molmo2	9B	`allenai/Molmo2-8B`
DeepSeekTiny	3B	`deepseek-ai/deepseek-vl2-tiny`
MiniCPM	1B	`openbmb/MiniCPM-V-4_5`

D Error Analysis

We conduct a large-scale error analysis of model responses, covering all graph-size partitions and both original and anonymized label settings. We use automated diagnostics to characterize both aggregate response biases and response-level failures. Specifically, we examine response bias, graph-constraint violations, yes-answer bias, ungrounded node predictions, answer-type mismatches, and generation and delivery failures. These categories are not mutually exclusive. Responses without a unique, valid ground-truth answer(e.g., MaximumClique) are excluded only from analyses that require a reference answer.

Prior graph and multimodal benchmarks have documented several related failure modes: models may exploit spurious correlations instead of the underlying graph structure (Wang et al., 2023), produce correct binary answers despite errors in their accompanying analyses (Babaiee et al., 2025), propagate graph-perception errors into downstream reasoning (Li et al., 2024), and violate explicit mathematical constraints (Xu et al., 2026). We describe these errors along with their frequency in our experiments in details:

Yes-answer bias. In VGH, the ground-truth labels for NodeCheckare exactly balanced between positive and negative instances. Nevertheless, BakLLaVA and LLaVA-7B answer "yes" in 97.3% and 96.6% of prompts, respectively. These models therefore produce apparently valid, parsable responses while providing almost no input-dependent discrimination.

Response bias. Several models repeatedly return a single numeric value irrespective of the input graph. We quantify this behavior using the response-bias ratio

$$R_{m,t,v} = \frac{\Pr(\hat{y} = v \mid m, t)}{\Pr(y = v \mid t)}, \tag{1}$$

where m is the model, t is the task, v is the dominant predicted value, $\hat{y}$ is the model prediction, and y is the ground truth. A large $R_{m,t,v}$ indicates that a value is predicted substantially more often than warranted by its prevalence in the data. Table 8 shows the value of the bias ratio among a subset of tasks and model pairs.

The strongest response concentration occurs for BakLLaVA on EdgeCount: the model predicts 10 in 55.5% of responses, whereas 10 accounts for only 0.9% of the corresponding ground-truth answers. Thus, the high prediction frequency cannot be explained by the ground-truth answer distribution alone. More generally, incorrect numeric predictions often concentrate on a small integer. For most models, the most frequent incorrect value accounts for 46–89% of the errors on the model's

Table 5: Detailed per-task performance on DOT visualization layout, expanding the category-level results in Table 6. P, A, and SA denote parsability, accuracy, and soft accuracy, respectively. All values are percentages.

Model	Graph Understanding										Graph Reasoning (Polynomial)												Graph Reasoning (NP-Complete)											
	Edge Count			Node Count			Node Degree			Node Check			Diameter			Highest Degree			Shortest Distance			Connected Components			Chordless Cycle			Maximum Clique			Maximum Independent Set			Minimum Vertex Cover			Traveling Salesman		
	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA	P	A	SA
Gemma-3n-E4B	100.0	12.2	62.7	100.0	39.8	86.6	100.0	38.9	60.9	100.0	95.6	95.6	100.0	16.5	43.6	100.0	29.2	70.5	100.0	17.1	27.6	100.0	30.6	36.5	100.0	8.8	100.0	100.0	2.3	30.6	100.0	3.2	14.8	100.0	6.5	29.3	99.9	0.0	6.2
DeepSeek-VL2-Tiny	100.0	17.3	54.6	100.0	30.2	81.3	99.7	42.0	59.1	100.0	93.0	93.0	100.0	26.4	54.3	100.0	30.2	58.9	100.0	33.4	46.1	98.7	22.2	28.2	99.0	27.4	67.7	95.8	5.7	14.6	97.2	1.5	6.6	94.8	3.0	15.1	98.2	0.0	0.0
bakLlava7b	100.0	0.5	36.8	100.0	9.3	65.9	97.9	14.7	38.9	100.0	55.3	55.3	100.0	12.2	38.1	95.4	11.5	59.2	100.0	16.0	31.7	100.0	25.4	32.6	100.0	21.9	100.0	100.0	0.0	0.3	99.9	0.1	0.6	100.0	0.0	0.0	100.0	0.0	0.0
gemma31b	98.7	28.9	68.4	98.7	37.1	84.8	98.9	41.6	60.8	98.8	96.7	96.7	98.6	23.7	45.3	98.7	32.7	69.4	98.7	39.3	49.9	98.7	69.6	75.6	98.7	10.5	65.3	98.7	3.5	16.8	98.7	10.0	20.8	98.7	7.0	17.8	97.9	0.0	2.7
gemma4-e4b	100.0	19.0	64.7	100.0	37.3	83.7	98.9	38.2	60.2	100.0	93.6	93.6	100.0	30.3	59.5	100.0	16.7	60.6	100.0	46.6	59.9	100.0	78.7	84.9	100.0	18.4	61.7	98.6	4.5	21.0	97.6	8.6	25.6	96.5	8.5	16.0	97.1	0.0	4.9
glm-v10b	100.0	32.6	68.6	100.0	58.4	94.6	96.4	50.6	69.5	100.0	98.8	98.8	99.8	25.3	51.0	100.0	27.9	68.9	100.0	56.5	65.1	100.0	82.1	86.9	99.9	12.2	67.0	100.0	8.5	20.0	100.0	9.6	15.6	100.0	5.1	13.3	100.0	0.0	5.6
internvl14b	87.8	22.9	64.2	76.7	31.9	82.3	98.1	41.7	61.1	98.8	98.0	98.0	98.6	41.3	65.3	98.3	24.0	66.0	99.9	46.6	56.8	99.9	84.4	90.1	98.7	11.1	65.6	98.7	7.5	24.6	73.7	7.6	12.5	98.7	7.5	11.1	88.5	0.0	4.4
internvl30b	95.5	30.7	65.4	98.7	57.3	93.9	95.1	31.1	47.3	98.8	98.4	98.4	98.7	38.4	58.9	98.7	37.4	75.7	98.7	57.9	68.2	98.7	83.4	88.7	85.8	11.4	35.8	98.6	6.0	15.1	98.6	12.0	23.3	96.9	8.0	9.9	98.6	0.0	1.1
internvl8b	98.7	28.8	70.2	98.7	51.4	92.8	98.2	51.3	70.7	99.2	99.2	99.2	98.7	30.7	55.8	98.7	23.8	63.3	100.0	53.1	63.3	98.7	83.2	88.8	98.7	8.5	65.4	98.7	5.9	17.7	98.7	8.7	25.0	98.7	6.6	14.0	98.7	0.0	3.9
llava13b	77.2	13.9	57.4	95.3	21.2	76.2	96.4	39.5	58.7	98.8	76.3	76.3	68.5	16.3	37.1	96.1	5.0	43.1	98.4	26.5	41.1	73.4	11.7	16.0	87.9	10.3	65.0	94.2	2.6	11.3	95.7	1.6	10.9	97.9	1.8	14.3	35.3	0.0	1.1
llava7b	100.0	15.6	55.4	100.0	25.9	70.1	99.3	41.2	59.3	100.0	53.7	53.7	100.0	22.5	46.2	89.7	10.4	48.1	98.9	30.2	42.7	100.0	19.2	22.8	99.9	14.2	67.2	100.0	1.8	10.2	100.0	1.0	7.4	100.0	1.1	10.1	99.7	0.0	0.0
minicpm	100.0	34.1	72.3	100.0	52.8	89.6	96.0	45.0	65.2	98.4	95.5	95.5	100.0	25.7	51.4	99.8	35.0	66.3	100.0	37.7	48.6	99.9	70.1	75.1	95.7	11.1	65.4	98.7	7.6	14.3	98.7	8.3	17.7	98.6	7.5	26.9	98.7	0.0	1.2
molmo7b	100.0	29.2	66.4	100.0	52.7	91.7	99.2	53.4	71.6	100.0	89.4	89.4	100.0	33.2	62.4	100.0	37.8	76.3	100.0	47.3	57.6	100.0	82.1	88.4	98.7	12.9	65.5	98.7	6.0	19.4	98.7	5.6	27.0	98.7	3.7	4.4	98.7	0.0	1.4
qwenvl32b	98.3	39.4	67.7	98.7	57.6	94.2	97.5	50.6	68.5	97.1	99.4	99.4	95.2	27.8	60.5	98.7	17.3	53.6	95.5	65.8	74.0	95.4	87.7	91.8	98.1	15.6	64.9	91.2	11.0	17.7	94.9	23.2	33.1	95.1	13.8	23.1	94.2	0.0	3.2
qwenvl8b	98.7	34.3	68.4	98.7	61.5	94.9	97.8	52.1	70.0	98.8	97.4	97.4	89.5	41.1	64.5	98.7	28.9	62.0	100.0	64.5	73.2	100.0	84.7	90.3	95.6	11.7	65.5	82.0	7.9	11.9	94.9	11.8	25.9	95.0	6.0	15.1	78.6	0.0	2.6

Table 6: Performance of VLMs on the graphs with images created by DOT visualization layout. For each task category, the macro-average value is reported across the tasks in that category.

Model	Graph Understanding			Graph Reasoning (Polynomial Time)			Graph Reasoning (NP-Complete)		
	Parsable	Accuracy	Soft Accuracy	Parsable	Accuracy	Soft Accuracy	Parsable	Accuracy	Soft Accuracy
Qwen3-VL-32B	97.9	61.8	82.5	96.2	49.7	70.0	94.7	12.7	28.4
Qwen3-VL-8B	98.5	61.4	82.7	97.1	54.8	72.5	89.2	7.5	24.2
InternVL-30B	97.0	54.4	76.3	98.7	54.3	72.9	95.7	7.5	17.0
InternVL-14B	90.3	48.6	76.4	99.2	49.1	69.5	90.9	6.7	23.5
InternVL-8B	98.7	57.7	83.2	99.0	47.7	67.8	98.7	5.9	25.2
Gemma3-27B	98.8	51.1	77.7	98.7	41.3	60.0	98.6	6.2	24.7
Gemma3-8B	100.0	46.6	76.4	100.0	23.3	44.6	100.0	4.2	36.2
Gemma4-8B	99.7	47.0	75.6	100.0	43.1	66.2	98.0	8.0	26.1
LLaVA-13B	91.9	37.7	67.1	84.1	14.9	34.3	82.2	3.3	20.5
LLaVA-7B	99.8	34.1	59.6	97.2	20.6	39.9	99.9	3.6	19.0
BakLLaVA	99.5	20.0	49.2	98.8	16.3	40.4	100.0	4.4	20.2
GLM	99.1	60.1	82.9	100.0	48.0	68.0	100.0	7.1	24.3
Molmo2	99.8	56.2	79.8	100.0	50.1	71.2	98.7	5.6	23.5
DeepSeekTiny	99.9	45.6	72.0	99.7	28.1	46.9	97.0	7.5	20.8
MiniCPM	98.6	56.9	80.7	99.9	42.1	60.4	98.1	6.9	25.1

Table 7: Effect of graph-visualization layout on the performance of VLMs on visual-understanding tasks with anonymized node labels. Soft and Hard denote soft accuracy and exact-match accuracy, respectively. All values are percentages; "–" indicates that results are unavailable. The best and second-best results in each column are shown in bold and underlined, respectively. The Acc column indicates accuracy and S ACC column indicates soft accuracy

Model	DOT Layout								NEATO Layout							
	NodeCheck		NodeCount		NodeDegree		EdgeCount		NodeCheck		NodeCount		NodeDegree		EdgeCount	
	S Acc	Acc	S Acc	Acc	S Acc	Acc	S Acc	Acc	S Acc	Acc	S Acc	Acc	S Acc	Acc	S Acc	Acc
Qwen3-VL-32B	99.3	99.3	**99.7**	**97.2**	74.0	42.7	36.0	0.0	99.7	99.7	99.5	93.7	86.7	60.3	37.8	0.0
Qwen3-VL-8B	96.0	96.0	99.6	94.6	71.8	42.4	44.9	0.2	**99.9**	**99.9**	**99.7**	**94.8**	**88.0**	**62.0**	42.2	0.1
InternVL-30B	92.7	92.7	99.2	87.7	47.8	23.5	42.2	0.4	**99.9**	**99.9**	97.3	70.4	80.5	51.0	**57.9**	0.4
InternVL-14B	97.6	97.6	99.1	92.4	67.0	37.5	46.4	0.0	**99.9**	**99.9**	98.0	76.1	76.4	47.6	49.3	0.0
InternVL-8B	99.9	99.9	97.0	69.5	**75.6**	**46.0**	**57.0**	0.3	99.8	99.8	96.9	63.2	82.8	53.6	57.0	0.3
Gemma3-27B	99.6	99.6	98.3	85.3	61.0	29.0	44.6	0.0	99.8	99.8	99.0	83.8	71.5	40.1	40.8	0.2
Gemma3-8B	99.2	99.2	95.7	52.2	58.2	23.8	52.3	0.3	99.8	99.8	96.2	49.0	65.9	32.8	41.4	0.2
Gemma4-8B	99.6	99.6	93.1	47.5	60.0	25.8	45.8	0.0	99.8	99.8	91.6	30.3	71.0	38.3	39.3	0.1
LLaVA-13B	96.9	96.9	79.6	3.5	48.7	20.3	39.3	0.6	99.2	99.2	76.6	3.8	49.7	22.6	33.8	0.3
LLaVA-7B	50.3	50.3	70.2	3.8	38.5	15.6	31.6	**1.4**	50.8	50.8	69.3	3.0	36.8	14.4	27.6	0.5
BakLLaVA	50.0	50.0	64.4	5.7	35.4	9.9	38.1	0.7	–	–	–	–	–	–	–	–
GLM	**100.0**	**100.0**	98.6	77.9	70.1	40.6	44.4	0.2	**99.9**	**99.9**	98.0	70.3	84.2	60.7	36.7	0.1
Molmo2	98.3	98.3	98.0	83.8	69.9	42.4	48.8	0.0	99.2	99.2	99.2	87.8	78.0	50.3	48.6	0.2
DeepSeekTiny	86.0	86.0	95.4	40.2	39.3	16.1	32.4	0.1	87.1	87.1	92.3	16.3	42.8	20.0	33.1	0.2
MiniCPM	99.9	99.9	98.4	80.0	67.4	34.9	56.0	0.5	99.8	99.8	97.8	75.3	81.7	**62.0**	51.6	**0.6**

most affected task. InternVL-14B and DeepSeekTiny show less concentrated incorrect-answer distributions, with corresponding peak shares of 34% and 23%, respectively.

Graph-constraint violations. We additionally test whether numeric predictions satisfy elementary graph constraints. For a graph with $n = |V|$ nodes, we mark a prediction as violating graph constraints if it violates a task-relevant bound, such as

$$\deg(v) < n, \qquad d(u, v) \leq n - 1, \qquad \mathrm{diam}(G) \leq n - 1, \tag{2}$$

Table 8: Representative cases of response bias. Pred. and GT columns denote the frequencies of the selected value in the model predictions and the ground-truth answers, respectively.

Task	Model	Value	Pred.	GT	Bias Ratio
EdgeCount	BakLLaVA	10	55.5%	0.9%	59×
NodeCount	BakLLaVA	10	51.5%	4.8%	11×
ShortestDistance	InternVL-30B	1	83.9%	11.8%	7×
ConnectedComponentCount	DeepSeekTiny	3	79.6%	5.5%	15×
HighestDegree	DeepSeekTiny	3	69.7%	5.0%	14×
ConnectedComponentCount	LLaVA-7B	10	26.6%	0.3%	87×
Diameter	Gemma4-8B	3	71.7%	21.3%	3×

or, for a simple undirected graph,

$$|E| \leq \frac{n(n-1)}{2}. \tag{3}$$

We similarly require predicted clique, independent-set, and vertex-cover sizes not to exceed n.

Across the evaluation, 82 model–task combinations contain at least five graph-constraint violations. SHORTESTDISTANCEis the most affected task: for 13 models, between 9% and 13% of parsed responses exceed the maximum feasible distance, corresponding to approximately 51,000 responses. The rate is nearly identical for original and anonymized labels (9.66% and 9.71%, respectively), indicating that these failures are not primarily caused by label anonymization. Instead, the models sometimes generate numerical magnitudes without grounding them in the graph. We observe that graph-constraint violations are more common in the small-graph partition than in the large-graph partition (13.9% versus 6.9%). Moderate violations are therefore partly affected by the tighter feasible range of small graphs. This size effect, however, cannot explain the extreme values.

Ungrounded node hallucinations. For tasks that require a set or sequence of nodes as output (e.g, MAXIMUMINDEPENDENTSET), we examine whether every node identifier extracted from the predicted answer matches a node label in the input graph. This error is observed only for Bak-LLaVA, which produces unmatched floating-point values in $[0, 1]$ in 100% of its successfully parsed list-valued responses under anonymized node labels and 96.3% under original labels, affecting approximately 49,600 responses in total. Because these values appear as elements of the predicted node sets or sequences but do not correspond to any node in the input graph, we classify them as ungrounded node hallucinations rather than incorrect selections among existing nodes.

Answer-type mismatches. We observe that some models fail to yield a response in the instructed format. However, the generated response is relevant to the task. For example, DeepSeekTiny responds to 35.2% of CHORDLESSCYCLE prompts with an explicit cycle rather than the requested numeric quantity, resulting in 4,622 responses without an extractable final answer. InternVL-14B and InternVL-30B exhibit a related but less frequent behavior on NODEDEGREE: in 4.1% and 3.0% of responses, respectively, they enumerate neighboring nodes but do not provide the requested degree. All other model and task combinations have mismatch rates below 0.5%. We do not classify a response as an answer-type mismatch when it contains a reasoning trace but terminates in the requested answer type.

Generation and delivery failures. We observe both inference-level failures and model-output failures that prevent task correctness from being evaluated. We classify a response as truncated when it reaches the configured output-length limit and ends before completion, typically without a final answer. LLaVA-13B refuses 4.3% of prompts, often citing insufficient image quality, and produces truncated responses in 1.6% of cases. InternVL-14B and InternVL-30B produce special-token-only responses in 4.6% and 1.9% of cases, respectively; these responses contain a control token such as <|im_end|> but no answer content. InternVL-30B additionally produces truncated responses in 1.1% of cases. Qwen3-VL-32B and Qwen3-VL-8B return empty responses in 3.0% and 1.9% of cases, respectively. LLaVA-7B, MiniCPM, DeepSeekTiny, and Gemma4-8B produce runaway generations that reach the output-length limit in approximately 1.6–3.1% of responses.

These observations highlight the significance of fine-grained evaluation in VGH: We report query success, parsability, and unscorable-response rates alongside task accuracy and report inference-level query failures separately. Moreover, our evaluators distinguish answer-type mismatches, ungrounded node predictions, and graph-constraint violations from incorrect but otherwise valid answers. Finally, we report task accuracy along with ground-truth distribution to avoid misinterpretation of constant response as highly accurate model.

E PROMPTS

Domain Information in Prompts

Social Network The following graph is a social network. In this network, nodes represent users and edges represent friendship connections.

Academic Network The following graph is an academic network. In this network, nodes represent authors and edges represent research collaborations.

Knowledge Graph The following graph is a knowledge graph. In this graph, nodes represent entities and edges represent the relationship between entities. The type of each relationship is specified by the edge label.

Flight Route Network The following graph is a flight route network. In this network, nodes represent airports and edges represent routes and airport distances (in kilometers).

Commutative Diagram The following graph is a commutative diagram. In a commutative diagram, nodes represent mathematical objects such as sets, groups, vector spaces, rings, topological spaces, etc. The edges (arrows) represent functions or morphisms between those objects.

Graph Understanding Tasks Prompts

NODECHECK
Does this graph have a node with the name {NODE LABEL PLACEHOLDER}? Answer only with Yes or No.

NODECOUNT
How many nodes are in this graph? Answer only with a number.

NODEDEGREE
What is the degree of {NODE LABEL PLACEHOLDER}? Answer only with a number.

EDGECOUNT
How many edges are in this graph? Answer only with a number.

Graph Reasoning Polynomial Tasks Prompts

HIGHESTDEGREE
What is the degree of a node that has the most connections, counting all edges regardless of direction? Answer only with a number.

CONNECTEDCOMPONENTCOUNT
Within a connected component of a graph, any node can be reached from any other node through the edges in the graph. Different connected components are isolated from each other. What is the number of connected components in this graph? Answer only with a number.

DIAMETER
The diameter of a graph is the maximum distance between any pair of nodes in the graph. Given a graph G = (V, E), this task entails finding (1) a shortest path between two arbitrary

vertices and (2) ensuring it is the maximum possible among all shortest paths. What is the diameter of this graph? Answer only with a number.

SHORTESTDISTANCE
The shortest path between two specified nodes is a path that minimizes the number of hops. What is the length of the shortest path between {NODE LABEL PLACEHOLDER} and {NODE LABEL PLACEHOLDER} in this graph? Answer only with a number. If there is no path between these nodes, reply with -1.

Graph Reasoning NP-Complete Tasks Prompts

MAXIMUMCLIQUE
Maximum Clique is the largest subset of nodes such that every pair of vertices in this subset is connected by an edge. Identify the clique with the maximum number of nodes in this graph. Present your answer in the following format: [NodeA, NodeB, NodeC, NodeD, ...]. Output only the nodes in the maximum clique. Do not explain.

MINIMUMVERTEXCOVER
Minimum Vertex Cover is the smallest subset of nodes such that every edge has at least one node from this subset. Identify the minimum vertex cover, in a given graph G = (V, E). Present your answer in the following format: [NodeA, NodeB, NodeC, NodeD, ...]. Output only the nodes in the maximum independent set. Do not explain.

MAXIMUMINDEPENDENTSET
Maximum independent set is the largest subset of nodes such that no two nodes in this subset are connected by an edge in a graph. Identify the maximum indepdendent, in a given graph G = (V, E). Present your answer in the following format: [NodeA, NodeB, NodeC, NodeD, ...]. Output only the nodes in the maximum independent set. Do not explain.

TRAVELINGSALESMAN
Traveling Salesman Problem in an undirected graph, finds the shortest possible path that visits each of the listed nodes exactly once and returns to the starting point. Please calculate the shortest tour and format your answer as follows and present your answer in the following format: [NodeA, NodeB, NodeC, NodeD, ...]. Output only the nodes in the tour. Do not explain.

CHORDLESSCYCLE Maximum chordless cycle in a graph is the cycle where no two nodes of the cycle are connected by an edge that is not part of the cycle itself. Determine the length of the maximum chordless cycle in this graph. Answer only with a number."