MVGRAPH: BENCHMARKING LVLMS ON EVIDENCE COMPOSITION ACROSS MULTI-VIEW VISUAL GRAPH

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce MVGRAPH, a benchmark for evaluating large vision-language models (LVLMs) on multi-graph reasoning where the evidence required by a query is distributed across multiple interrelated visual graph views. Although such evidence commonly resides in source-specific graphs, temporal snapshots, or local observations, existing graph reasoning benchmarks remain confined to a single self-contained structure, assessing only graph perception and reasoning while leaving the alignment and composition of evidence across views underexplored. MVGRAPH comprises 8k instances spanning 8 *graph-theoretic* tasks, abstracted from two multi-graph scenarios (partial and temporal graph views). Each problem is programmatically constructed and verified to be unsolvable from a single view. Beyond end-to-end evaluation, we decompose multi-graph reasoning into four complementary capabilities—*Perception*, *Alignment*, *Composition*, and *Reasoning*—and design 29 diagnostic tasks with 64k instances to probe each one, along with golden intermediate representations for evaluating the intermediate process. Across 15 LVLMs, we reveal the difficulty of multi-graph reasoning and identify cross-view edge alignment and relation-level composition as prominent bottlenecks through stage-oriented diagnostics. Meanwhile, temporal constraints and increasing view counts expose further capability boundaries. Code and data are available at `https://anonymous.4open.science/r/MVGraph-8C37`.

1 INTRODUCTION

Graphs provide a natural abstraction of relational structure in molecular property modeling (Gilmer et al., 2017), web-scale recommendation (Ying et al., 2018), traffic forecasting (Li et al., 2018), and knowledge-base reasoning (Xu et al., 2026b). Although these applications are typically formalized as computation over a single graph, the information required by a query is often distributed across multiple graphs rather than contained in one of them. It may reside in source- or relation-specific graphs that capture complementary aspects of the same system, in temporal snapshots that record how topology and attributes evolve (Kazemi et al., 2020), or in local observations that each cover only a subset of the relevant nodes and edges. Determining the answer therefore requires establishing cross-graph correspondences and combining complementary information across graphs, rather than reasoning over a single complete graph.

Large language models (LLMs) and large vision-language models (LVLMs) have recently shown strong performance on graph-structured tasks. LLMs have achieved strong results on text-serialized graphs, including connectivity, path finding, and combinatorial optimization (Wang et al., 2023; Chen et al., 2024). LVLMs have further advanced this line of work by rendering graphs as visual graphs, which provide a unified interface that naturally captures relational structure while remaining compatible with multimodal graphs, visually grounded knowledge bases, and other graph types that are difficult to serialize as text (Das et al., 2024; Wei et al., 2024; Deng et al., 2024). Existing evaluations, however, remain largely confined to single-graph settings, where a model need only perceive one graph and reason over it. In contrast, a multi-graph query requires the model to parse the structure of each view, match entities or states that recur across graphs, integrate complementary evidence, and only then infer the answer. This leaves a critical research gap: *Existing evaluations do not establish whether current LVLMs can recover and compose structural evidence distributed across multiple interrelated graph views when no individual view suffices.*

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.

To address this gap, we introduce **MVGRAPH**, a benchmark for evaluating LVLMs' graph reasoning across multiple interrelated visual graph views. As illustrated in Figure 1, unlike prior benchmarks that primarily evaluate single-graph reasoning—perceiving a self-contained structure and deriving an answer from it, MVGRAPH targets multi-view reasoning, where the answer further depends on aligning recurring entities and combining complementary evidence across views. Specifically, MVGRAPH contains 72,000 instances spanning two scenarios and 37 graph-theoretic and diagnostic tasks, with controlled variation in graph number and size, alignment ambiguity, cross-graph dependency depth, graph heterogeneity, and irrelevant evidence. Each instance is programmatically constructed and validated such that no individual view contains sufficient evidence to determine the answer, making task success dependent on evidence distributed across views.

To make the constituent capabilities of multi-view graph reasoning directly measurable rather than inferred from final accuracy, MVGRAPH introduces a capability-oriented evaluation schema. It decomposes multi-graph reasoning into four complementary capabilities: **Perception**, **Alignment**, **Composition**, and **Reasoning**. *Perception* captures the structure and content of each graph; *Alignment* identifies corresponding entities, relations, or states across graphs; *Composition* organizes complementary cross-graph evidence into a coherent representation; and *Reasoning* derives the answer from the integrated evidence. For each capability, we design targeted evaluation tasks together with a progressive evaluation that supplies verified outputs from preceding capabilities, enabling direct capability assessment and measurement of the resulting performance recovery. MVGRAPH thus evaluates not only overall multi-view graph reasoning, but also each constituent capability and the capability that most constrains model performance.

Across 15 LVLMs, our end-to-end evaluation shows that graph-theoretic reasoning over multiple visual views remains challenging. Through further stage-oriented diagnostics on seven LVLMs, we find that **cross-view edge alignment and relation-level composition remain significant weaknesses, despite stronger performance in identifying recurring nodes and relevant views**. Providing explicit structural evidence substantially improves performance on several partial-view graph-theoretic tasks, suggesting that **difficulties in processing distributed visual evidence** contribute to poor end-to-end performance. Tasks based on temporal sequences nevertheless remain challenging even with prepared structures, revealing an additional limitation in reasoning across snapshots. Finally, composition performance also deteriorates markedly for several LVLMs as the number of views increases. These findings establish MVGRAPH not only as a new evaluation resource, but also as a means to uncover capability differences obscured by existing single-graph evaluations.

Our contributions are as follows:

- **A comprehensive multi-graph benchmark.** We construct MVGRAPH, spanning two scenarios and 37 tasks with controlled complexity, genuine cross-graph dependencies, and gold intermediate representations.
- **A capability-oriented diagnostic schema.** We organize multi-graph reasoning into Perception, Alignment, Composition, and Reasoning, and combine capability-level metrics with progressive evaluation to identify model bottlenecks.
- **Empirical findings and validated measurements.** Across 15 LVLMs, we reveal the difficulty of multi-graph reasoning and identify cross-view edge alignment and relation-level composition as prominent bottlenecks through stage-oriented diagnostics. We further characterize limitations under temporal constraints and increasing view counts, while programmatic checks and view ablations provide evidence for answer correctness and multi-view dependence.

2 RELATED WORK

Visual Graph Benchmarking for LVLMs. Recent studies have begun to evaluate LVLMs on visual graph understanding and reasoning. VisionGraph (Li et al., 2024) evaluates visual graph-theoretic problems, while VGCURE (Zhu et al., 2025) broadens the evaluation to fundamental graph perception and reasoning. VisGraphVar (Sartori et al., 2025) studies robustness to visual variations, and DiGraphHal-Bench (Fan et al., 2026) focuses on hallucination over complex directed graphs. Some benchmarks also place multiple graphs within a single image, where IsoBench (Fu et al., 2024) includes graph isomorphism within a broader evaluation of visual and textual representations, Visual Graph Arena (Babaiee et al., 2025) tests isomorphism across different layouts, and VisGraphVar compares drawings of labeled graphs under visual style changes. More recently, GraphVerse (Sun

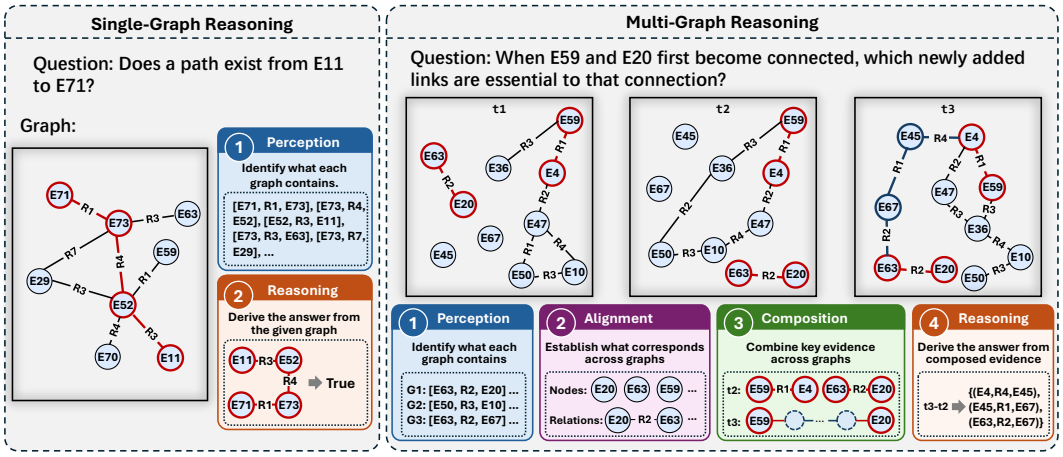


Figure 1: Illustration of evidence-fragmented visual graph reasoning in MVGRAPH, where three views provide complementary path segments.

et al., 2026) extends visual graph evaluation to paired images through graph edit distance and maximum common subgraph tasks. However, these studies do not systematically evaluate graph-theoretic problem solving when the necessary structure must be assembled from multiple complementary partial views or interpreted across ordered temporal views, together with diagnostics of the capabilities involved in that process.

Multi-Image Visual Reasoning. Recent benchmarks have extended LVLM evaluation from single-image understanding to reasoning over multiple images. MIRB (Zhao et al., 2024) and MUIRBENCH (Wang et al., 2025a) evaluate a broad range of multi-image relations, including comparison, co-reference, spatial understanding, temporal reasoning, and multi-hop inference, while MMIU (Meng et al., 2025) and MIBENCH (Liu et al., 2024) further expand task diversity and evaluation scale across heterogeneous multi-image scenarios. MILEBENCH (Song et al., 2024) studies long-context multimodal understanding as the number of input images increases, and reasoning-oriented benchmarks such as REMI (Kazemi et al., 2024) and MMRB (Cheng et al., 2025) emphasize multi-step inference over visual evidence. More recently, VCBENCH (Wang et al., 2025b) and OMIBENCH (Chen et al., 2026) construct problems in which complementary evidence is distributed across multiple images, reducing the possibility of answering from a single image alone. These benchmarks motivate evaluating whether models genuinely use multiple images, but fail to target the graph-specific demands of recovering local topology, tracking recurring or changing graph elements, composing task-relevant structures, and solving graph-theoretic problems. MVGRAPH studies these demands through end-to-end graph-theoretic challenges and stage-oriented tasks probing perception, alignment, composition, and reasoning when no individual view suffices.

3 MVGRAPH: BENCHMARK DESIGN AND CONSTRUCTION

MVGRAPH evaluates how LVLMs reason over multiple interrelated visual graph views. As shown in Figure 2, it combines end-to-end *graph-theoretic* challenges with stage-oriented diagnostic tasks that probe the *perception*, *alignment*, *composition*, and *reasoning* capabilities required to solve them. Both components are instantiated in two settings. In *partial graph views*, images expose different portions of an underlying graph, and task-relevant structure must be assembled across views. In *temporal graph views*, images depict successive graph states, and structural evidence must be interpreted in temporal order rather than merged indiscriminately.

3.1 TASK DESIGN

End-to-End Graph-Theoretic Challenges. The primary challenge suite consists of 8 graph-theoretic tasks across four problem families, including *topological sorting*, *maximum flow*, *bipartite matching*, and *Hamiltonian path*. In partial graph views, models must assemble the relevant

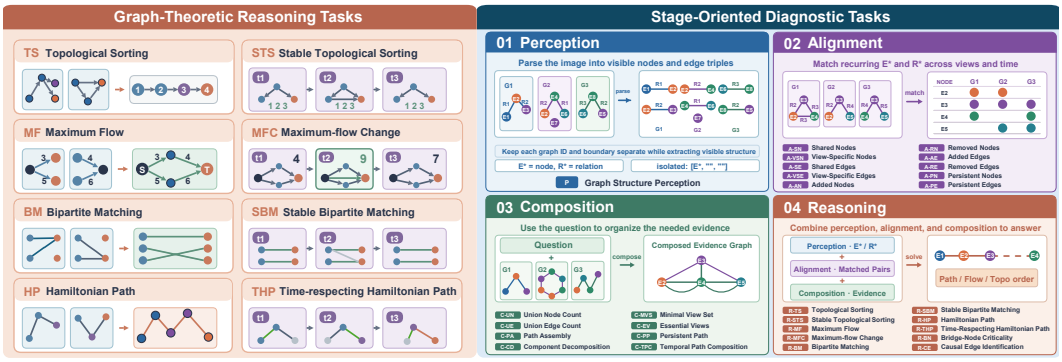


Figure 2: Overview of MVGRAPH. *Left:* End-to-end graph-theoretic tasks span partial and temporal graph views. *Right:* Stage-oriented diagnostic tasks probe perception, alignment, composition, and reasoning.

structure across images before solving the corresponding problem. In temporal graph views, they must account for structural evolution: producing topological orders while limiting the reordering of persistent nodes, comparing maximum-flow values between successive states, finding maximum matchings while limiting changes to persistent matches, or constructing a time-respecting Hamiltonian path. These tasks are evaluated end-to-end from graph images and cannot, particularly in the temporal setting, be reduced to reasoning over an indiscriminate union of observed edges.

Stage-Oriented Diagnostic Tasks. To examine capabilities underlying end-to-end performance, we organize diagnostic evaluation around perception, alignment, composition, and reasoning. (I) In *Perception*, models receive multiple graph images and produce a separate textual representation of the nodes and edge triples in each view. (II) *Alignment* tasks probe the identification of shared, view-specific, added, removed, and persistent graph elements across views. (III) *Composition*-oriented tasks examine whether models can organize task-relevant evidence, including assembled paths, component subgraphs, necessary view sets, and temporal path structures. (IV) *Reasoning* is assessed through structure-prepared versions of the eight graph-theoretic challenges, together with bridge-node criticality and causal edge identification. For the perception, alignment, and composition diagnostics, models receive only the graph images and the corresponding query. For the reasoning diagnostic, we provide task-relevant graph structure in textual form, allowing us to focus on structure reasoning without requiring visual evidence extraction.

3.2 INSTANCE CONSTRUCTION

Graph Sources and View Generation. We construct instances from procedurally generated graphs and graph topologies derived from SmallPedia. Procedural generation provides control over task-specific structural properties, while SmallPedia-derived topologies increase variation in the underlying graph structure. For each instance, we select an underlying graph or sequence of graph states that satisfies the task constraints and generate multiple interrelated views from it. Partial-view instances expose overlapping portions of an underlying graph, whereas temporal instances depict ordered states with changes in nodes or edges. Entity identifiers remain consistent across views, but each image is laid out independently so that corresponding entities cannot be identified by position alone. We use semantically neutral `E*` identifiers for nodes and `R*` identifiers for edge relations, where maximum-flow tasks additionally display numerical edge capacities.

Question and Answer Generation. For each task, we select graph instances satisfying its structural constraints and instantiate a question template with the relevant entities, view indices, and time steps. Reference answers are computed from the underlying graph records using task-specific algorithms. Diagnostic answers include graph-element sets, counts, edge-triple paths, component subgraphs, and view indices, while temporal path composition additionally requires the two component subgraphs before a connectivity transition and a path after it. The graph-theoretic tasks require outputs such as topological orders, maximum-flow values, maximum matchings, and Hamiltonian

Table 1: Statistics of MVGRAPH.

Task group	# Tasks	# Instances	Views	Avg. views	Avg. nodes	Avg. edges
Graph-Theoretic	8	8,000	2–4	3.06	14.77	34.03
Perception	1	36,000	2–8	4.46	21.47	26.93
Alignment	10	10,000	2–8	4.89	24.07	28.49
Composition	8	8,000	2–8	4.81	25.33	24.67
Reasoning	10	10,000	2–8	3.43	16.19	30.98
Total	**37**	**72,000**	**2–8**	–	–	–

paths, with time-indexed solutions where appropriate. For tasks that may admit multiple valid solutions, we use task-specific programmatic verifiers to check whether each predicted solution is feasible or optimal, thereby accepting any valid solution rather than requiring an exact match to a single reference answer.

3.3 VALIDATION AND DATASET STATISTICS

Validation. We validate each instance against the structural prerequisites of its task, including acyclicity, valid edge capacities, or bipartiteness. We then check that the reference answer computed from the underlying graph structures is consistent with the question and the selected views. For tasks with multiple valid solutions, we verify that the stored answer satisfies the task constraints and assess model outputs by the same constraints rather than by exact agreement with one reference answer. We additionally apply task-specific evidence-necessity checks to exclude instances that can be answered from a single view or without a required view. Finally, we check that the rendered images preserve the intended graph elements and that their annotations are legible.

Dataset Statistics. MVGRAPH comprises 37 evaluation tasks, including 8 end-to-end graph-theoretic challenges and 29 stage-oriented diagnostic tasks covering perception, alignment, composition, and reasoning. Across these evaluation protocols, the benchmark contains 72,000 task-instance records, each involving 2–8 interrelated visual graph views. The end-to-end challenges use 2–4 views, while the diagnostic tasks extend the evaluation to as many as 8 views. Table 1 summarizes the statistics of MVGRAPH. More details are available in Appendix A.

4 EXPERIMENTAL SETUP

Models. We evaluate 15 distinct LVLMs on the eight end-to-end graph-theoretic tasks with thinking or no-thinking mode. The stage-oriented diagnostics are conducted only with the seven thinking models, comprising four locally deployed models, including Qwen3.5-9B (Team, 2026), Gemma-4-12B-it (Team et al., 2026), GLM-4.6V-Flash (Hong et al., 2025), and Ovis2.5-9B (Lu et al., 2025) and three API-accessed models, including Qwen3.8-Flash (Qiu et al., 2026), GLM-5.3-Flash (Zeng et al., 2026), and DeepSeek-Flash (Xu et al., 2026a). All local inference is performed using vLLM.

Evaluation Protocol. For the end-to-end graph-theoretic tasks, models receive multiple graph images, followed by the task question and output-format instruction. The perception, alignment, and composition diagnostics likewise use visual graph inputs. In perception, multiple images are presented together and the model reconstructs the nodes and edge triples of each view separately. For the reasoning diagnostics, models instead receive task-prepared textual graph structures and composition evidence, allowing us to assess reasoning when the relevant structure has been made explicit. All evaluations are zero-shot.

Metrics. For perception, we report exact-match accuracy and node and edge F1 by comparing the reconstructed structure of each view with its ground-truth graph. For all other tasks, we report exact-match based accuracy, assigning no partial credit. Set-valued answers are compared after order-insensitive normalization, whereas paths and temporal outputs must preserve their required order. For graph-theoretic tasks with multiple valid solutions, task-specific programmatic verifiers determine whether a prediction is valid and, where required, optimal.

Table 2: Accuracy (%) on end-to-end graph-theoretic tasks of MVGRAPH in thinking mode. † denotes models accessed via API. The best result in each column within each model group is bolded.

Models	Partial graph views				Temporal graph views				Avg.
	TS	MF	BM	HP	STS	MFC	SBM	THP	
Qwen3.5-9B	**0.1**	0.3	36.9	40.0	0.0	30.2	0.2	0.4	13.5
Gemma-4-12B-it	0.0	**5.5**	**60.6**	**43.1**	**0.1**	45.3	**0.4**	**8.7**	**20.5**
GLM-4.6V-Flash	0.0	0.7	7.2	4.7	0.0	**46.5**	0.0	0.0	7.4
Ovis2.5-9B	0.0	0.2	18.5	2.3	0.0	41.3	0.0	0.2	7.8
Qwen3.8-Flash†	**35.0**	**54.0**	**97.0**	77.0	8.0	**59.0**	8.0	65.0	50.4
GLM-5.3-Flash†	33.0	41.0	95.0	**84.0**	**12.0**	**59.0**	**16.0**	**79.0**	**52.4**
DeepSeek-Flash†	0.0	15.0	62.0	50.0	0.0	5.0	0.0	7.0	17.4

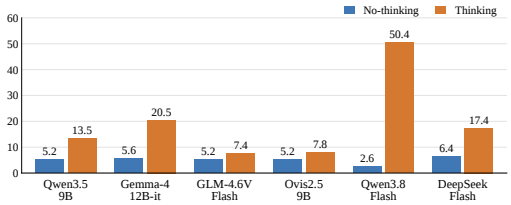


Figure 3: Average accuracy on graph-theoretic tasks with and without thinking.

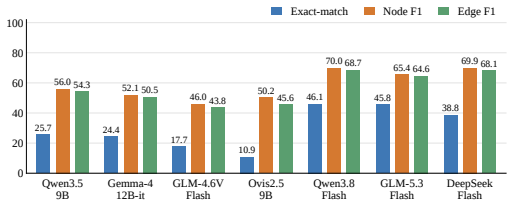


Figure 4: Perception performance of LVLMs in thinking mode.

5 RESULTS AND ANALYSIS

We organize our study around four research questions: (RQ1) how well current LVLMs reason across multiple interrelated graphs; (RQ2) which stage limits performance, localized through progressive oracle interventions; (RQ3) how performance scales with graph views; and (RQ4) whether the benchmark reliably measures the intended capabilities.

5.1 END-TO-END GRAPH-THEORETIC PERFORMANCE

Main results. Table 2 reports end-to-end performance on the eight graph-theoretic tasks. Overall, **performance remains far from saturated**. GLM-5.3-Flash achieves the highest reported average, closely followed by Qwen3.8-Flash, while Gemma-4-12B-it leads the locally deployed models. DeepSeek-Flash trails the other two API-accessed models, suggesting that substantially greater model scale alone does not guarantee reliable multi-graph reasoning. **Performance also differs across the two multi-graph settings**. The two leading LVLMs generally perform better on partial graph views than on temporal graph views. In particular, their strong results on Bipartite Matching (BM) do not carry over to Stable Bipartite Matching (SBM) across snapshots, and Stable Topological Sorting (STS) remains difficult. Maximum-flow Change (MFC) is an exception to this pattern, although its comparatively high scores warrant caution because the task asks models to identify a transition rather than report a numerical flow value. Taken together, these results demonstrate substantial variation across models and graph settings, while leaving the sources of their errors unresolved. More results and analysis are provided in the Table 14 and Appendix C.1.

Effect of thinking. Figure 3 compares the six models evaluated in both inference settings. Without thinking, their average accuracies cluster between 2.6% and 6.4%. Thinking improves every model, but the gains vary considerably. Qwen3.8-Flash shows the largest increase, from 2.6% to 50.4%, whereas GLM-4.6V-Flash and Ovis2.5-9B improve only modestly. Thus, additional test-time reasoning can substantially benefit multi-graph problem solving, but its effect is highly model-dependent and does not eliminate the difficulty of the benchmark.

Table 3: Accuracy (%) of thinking-mode LVLMs on the ten alignment tasks of MVGRAPH. † denotes models accessed via API. The best result in each column is bolded.

Models	Partial				Temporal						Avg.
	A-SN	A-VSN	A-SE	A-VSE	A-AN	A-RN	A-AE	A-RE	A-PN	A-PE	
Qwen3.5-9B	82.1	94.0	55.2	36.0	98.1	92.1	29.8	35.5	99.6	75.5	69.8
Gemma-4-12B-it	81.3	91.0	57.1	29.8	98.4	91.9	46.5	44.9	95.0	80.6	71.7
GLM-4.6V-Flash	97.9	68.0	58.8	11.0	81.7	59.9	18.4	19.1	96.9	62.9	57.5
Ovis2.5-9B	69.4	65.7	53.4	4.0	86.1	72.0	8.3	12.6	92.2	65.9	53.0
Qwen3.8-Flash†	**99.0**	**99.5**	74.5	**57.5**	99.5	**99.0**	63.0	65.0	**100.0**	**89.5**	84.7
GLM-5.3-Flash†	97.5	99.0	**85.5**	54.0	99.5	97.0	65.0	**65.5**	**100.0**	88.5	**85.2**
DeepSeek-Flash†	98.5	97.5	31.5	28.0	**100.0**	97.0	45.5	56.5	**100.0**	79.5	73.4

Table 4: Accuracy (%) of thinking-mode LVLMs on the eight composition tasks in MVGRAPH. † denotes models accessed via API. The best result in each column is bolded.

Models	Partial						Temporal		Avg.
	C-UN	C-UE	C-PA	C-CD	C-MVS	C-EV	C-PP	C-TPC	
Qwen3.5-9B	94.1	33.7	95.0	77.5	99.1	99.2	98.0	58.5	81.9
Gemma-4-12B-it	92.0	43.1	91.0	72.0	91.3	90.7	98.5	36.0	76.8
GLM-4.6V-Flash	68.0	21.9	58.5	32.5	70.4	69.3	96.5	6.5	53.0
Ovis2.5-9B	60.6	16.8	40.0	40.0	66.3	63.0	96.5	1.0	48.0
Qwen3.8-Flash†	**99.0**	67.0	**98.0**	**89.5**	99.5	99.5	**100.0**	**87.0**	**92.4**
GLM-5.3-Flash†	98.0	**73.0**	**98.0**	83.5	**100.0**	**100.0**	**100.0**	73.5	90.8
DeepSeek-Flash†	95.0	26.5	95.0	85.0	92.0	89.5	99.5	81.5	83.0

5.2 DIAGNOSING THE FOUR CAPABILITIES

End-to-end results do not reveal which capabilities constrain multi-graph reasoning. We therefore evaluate seven representitive LVLMs on diagnostic tasks targeting perception, alignment, composition, and reasoning. We use thinking mode throughout to assess these capabilities under the stronger inference setting observed in our end-to-end experiments.

Perception. Figure 4 shows that complete reconstruction of multiple graph views remains difficult even in thinking mode, with no model exceeding 47% exact-match accuracy. Node F1 consistently exceeds Edge F1, suggesting that extracting edge triples poses an additional challenge. **The gap between F1 and exact match indicates that LVLMs often recover part of the structure without reconstructing every view completely.** These results establish the difficulty of visual structure extraction before we examine the cross-view alignment and composition capabilities required specifically for multi-graph reasoning.

Alignment. Table 3 reveals a pronounced gap between node-level (A-SN, A-VSN, A-AN, A-RN, and A-PN) and edge-level (A-SE, A-VSE, A-AE, A-RE, and A-PE) alignment across all evaluated models. Shared, view-specific, added, removed, and persistent nodes are consistently identified more accurately than their edge counterparts. The gap is especially evident for view-specific edges, on which even the strongest model remains below 60% accuracy. This difficulty persists despite consistent entity identifiers across views, suggesting that **tracking how relations correspond or change across views is substantially harder than aligning nodes. Since such edge-level alignment is essential for integrating structural evidence, its weakness may limit LVLMs' multi-graph reasoning performance.**

Composition. Table 4 shows that composition difficulty depends strongly on the structural evidence that must be integrated. Models perform well on Minimal View Set (C-MVS) and Essential Views (C-EV), indicating that they can often identify which views contribute to a connection. Yet this ability does not consistently extend to composing the relations within those views. Union Edge Count (C-UE) is substantially less accurate than Union Node Count (C-UN) for every model, with even the strongest models falling below 75% on C-UE. A similar contrast appears in the temporal setting. Persistent Path (C-PP) is solved reliably, whereas Temporal Path Composition (C-TPC),

Table 5: Accuracy (%) of thinking-mode LVLMs on the ten structure-prepared reasoning tasks in MVGRAPH. Models receive textualized structural evidence instead of graph images. † denotes models accessed via API. The best result in each column is bolded.

Models	Partial					Temporal					Avg.
	R-TS	R-MF	R-BM	R-HP	R-BN	R-STS	R-MFC	R-SBM	R-THP	R-CE	
Qwen3.5-9B	72.0	62.0	79.0	87.0	93.0	0.0	46.0	0.0	1.0	97.0	53.7
Gemma-4-12B-it	91.0	41.0	71.0	89.0	97.0	0.0	67.0	5.0	44.0	96.0	60.1
GLM-4.6V-Flash	0.0	2.0	9.0	3.0	50.5	0.0	52.0	0.0	0.0	65.5	18.2
Ovis2.5-9B	2.0	14.0	25.0	5.0	56.5	0.0	34.0	0.0	0.0	62.5	19.9
Qwen3.8-Flash†	93.0	73.0	93.0	86.0	87.0	16.0	70.0	28.0	65.0	69.5	68.1
GLM-5.3-Flash†	95.0	87.0	95.0	98.0	98.5	**44.0**	89.0	**55.0**	**84.0**	96.5	**84.2**
DeepSeek-Flash†	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	19.0	**99.0**	32.0	66.0	**100.0**	81.6

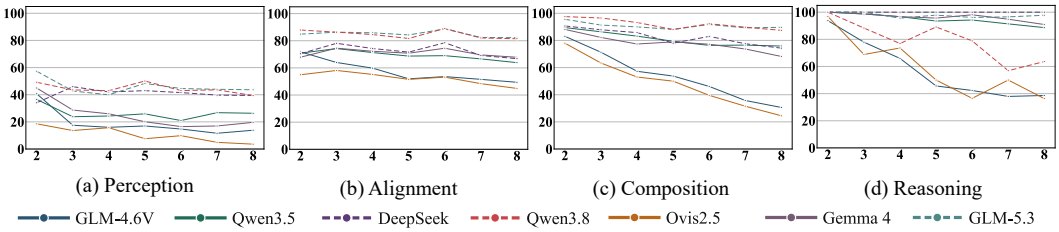


Figure 5: Diagnostic performance with thinking-mode across various number of graph views. The reasoning includes only Bridge-Node Criticality (R-BN) and Causal Edge Identification (R-CE).

which requires reconstructing components before a transition and a connecting path afterward, is markedly more difficult. **Therefore, selecting relevant views is easier than faithfully combining their edge-level evidence into a complete cross-view structure, which may in turn limit graph-theoretic reasoning over distributed evidence.**

Reasoning. Unlike the preceding diagnostics, this condition provides textualized graph structures with prepared composition evidence. Table 5 shows that several LVLMs can solve the partial-view graph-theoretic challenges when the relevant structure is made explicit. DeepSeek-Flash, for example, solves all such challenges in this condition despite performing substantially worse on their image-based counterparts. This contrast indicates that **weak end-to-end performance cannot be attributed to graph-theoretic inference alone** and is consistent with the cross-view alignment and composition difficulties identified above. Furthermore, Stable Topological Sorting (R-STS) and Stable Bipartite Matching (R-SBM) remain difficult even with prepared structures, **revealing an additional limitation in reasoning over solutions that must remain consistent across time.**

Summary. The four diagnostics show that multi-graph reasoning is not limited by visual perception alone. Models often identify recurring nodes and relevant views, but **struggle to align edges and compose their relations into a coherent cross-view structure.** Temporal tasks place further demands on this capability by **requiring models to align which relations persist or change across snapshots.** Although explicit structural evidence improves performance on several graph-theoretic tasks, the remaining difficulty of temporally coupled problems shows that reasoning across time is also challenging.

5.3 SCALING WITH THE NUMBER OF VIEWS.

Figure 5 examines diagnostic performance as the number of graph views increases. Perception exact match generally declines when models must reconstruct more views jointly. The most pronounced degradation appears in composition, where several locally deployed models lose substantial accuracy as structural evidence is spread across more views, while the API-accessed models remain comparatively stable. Aggregate alignment performance changes less consistently with view count, although the task-level results above reveal persistent difficulty in aligning edges. These patterns

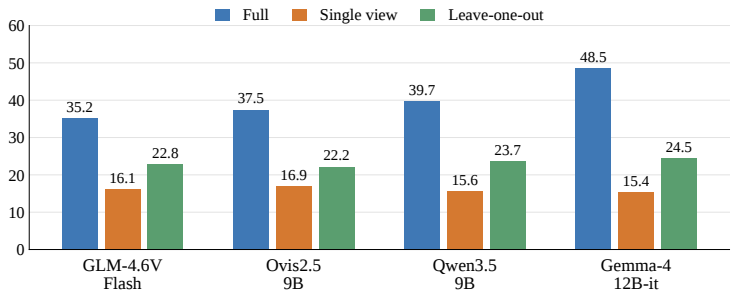


Figure 6: Effect of view availability on MVGRAPH performance, where *Full* uses all graph views, *Single View* retains one randomly selected view, and *Leave-One-Out* removes that same view. Bars show mean accuracy on the sampled instances.

suggest that faithfully integrating relation-level evidence across many views is a particularly important capacity boundary for current LVLMs.

5.4 BENCHMARK VALIDATION

Programmatic verification. MVGRAPH is constructed using task-specific graph-generation programs and question templates, rather than manually assigned answers. Each generated instance is checked against the structural constraints of its task, including valid edge endpoints and applicable properties such as acyclicity or bipartite structure. Its reference answer is then computed from the underlying graph structures and verified by a task-specific program. For tasks with multiple valid solutions, the verifier checks whether a candidate satisfies the required structural constraints and, where applicable, achieves the optimum, rather than requiring it to match a single reference string. These checks ensure that the stored answers are correct with respect to the generated graphs and the specified task definitions.

View-ablation protocol. We further examine whether answering the questions benefits from access to multiple graph views. Specifically, we randomly select a subset of MVGRAPH instances and evaluate four LVLMs under three paired input conditions without thinking. *Full* presents every graph view. For each instance, we randomly choose one view I_j; *Single View* presents only I_j, whereas *Leave-One-Out* presents every view except I_j. The selected index is held fixed across models, and the question and evaluation procedure remain unchanged across the three conditions.

Results. Figure 6 shows a consistent ordering across all test LVLMs, with *Full* outperforming *Leave-One-Out*, which in turn outperforms *Single View*. Accuracy drops sharply when only one graph is available and remains substantially below *Full* even when just one view is removed. For the strongest model, omitting a view nearly halves its accuracy. Moreover, model performance is much more similar under the reduced-view conditions than under *Full*, suggesting that their ability to exploit distributed evidence becomes most apparent when the complete set of views is available. These results provide behavioral evidence of multi-view dependence at the dataset level, complementing the programmatic checks of answer correctness.

6 CONCLUSION

We introduced MVGRAPH to evaluate reasoning over structural evidence distributed across multiple interrelated visual graph views. Its graph-theoretic challenges and stage-oriented diagnostics reveal that current LVLMs struggle not only to perceive graph structure, but also to align edge relations and compose evidence across views. Providing explicit structural information improves several partial-view tasks, yet temporally coupled reasoning remains difficult, and composition performance declines for several models as the number of views grows. These findings highlight relation-level alignment, faithful evidence composition, and temporal reasoning as important directions for improving multi-graph understanding.

AI USE STATEMENT.

We used ChatGPT to assist with language polishing and improving the clarity of the manuscript. The authors reviewed and take responsibility for the final content.

REFERENCES

Zahra Babaiee, Peyman Kiasari, Daniela Rus, and Radu Grosu. Visual graph arena: Evaluating visual conceptualization of vision and multimodal large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 2081–2113, 13–19 Jul 2025.

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.

Qiguang Chen, Chengyu Luan, Jiajun Wu, Qiming Yu, Yi Yang, Yizhuo Li, Jingqi Tong, Xiachong Feng, Libo Qin, and Wanxiang Che. OMIBench: Benchmarking olympiad-level multi-image reasoning in large vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45100–45135, 2026.

Ziming Cheng, Binrui Xu, Lisheng Gong, Zuhe Song, Tianshuo Zhou, Shiqi Zhong, Siyu Ren, Mingxiang Chen, Xiangchao Meng, Yuxin Zhang, et al. Evaluating mllms with multimodal multi-image reasoning benchmark. *arXiv preprint arXiv:2506.04280*, 2025.

Debarati Das, Ishaan Gupta, Jaideep Srivastava, and Dongyeop Kang. Which modality should I use - text, motif, or image? : Understanding graphs with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 503–519, June 2024.

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

Yixin Fan, Zhao He, Yuxin Hou, Changhua Zhou, Zihao Liu, Peng Wang, Chenglong Lu, Xu Zhang, and Wei Wang. Digraphhal-bench: Evaluating multimodal large language models on complex directed graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30885–30894, June 2026.

Deqing Fu, Ruohao Guo, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. Isobench: Benchmarking multimodal foundation models on isomorphic representations. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=KZd1EErRJ1.

Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, et al. Tgb 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. *Advances in neural information processing systems*, 37:140199–140229, 2024.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, 2017.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Mehran Kazemi, Nishanth Dikkala, Ankit Anand, Petar Devic, Ishita Dasgupta, Fangyu Liu, Bahare Fatemi, Pranjal Awasthi, Dee Guo, Sreenivas Gollapudi, et al. Remi: A dataset for reasoning with multiple images. *Advances in Neural Information Processing Systems*, 37:60088–60109, 2024.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020. URL http://jmlr.org/papers/v21/19-447.html.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

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. In *Forty-first International Conference on Machine Learning*, 2024.

Haowei Liu, Xi Zhang, Haiyang Xu, Yaya Shi, Chaoya Jiang, Ming Yan, Ji Zhang, Fei Huang, Chunfeng Yuan, Bing Li, and Weiming Hu. MIBench: Evaluating multimodal large language models over multiple images. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22417–22428, 2024.

Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, et al. Ovis2. 5 technical report. *arXiv preprint arXiv:2508.11737*, 2025.

Fanqing Meng, Jin Wang, Chuanhao Li, Quanfeng Lu, Hao Tian, Tianshuo Yang, Jiaqi Liao, Xizhou Zhu, Jifeng Dai, Yu Qiao, et al. Mmiu: Multimodal multi-image understanding for evaluating large vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 38405–38453, 2025.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, et al. On the design of qwen3. 8-next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Camilo Chacón Sartori, Christian Blum, and Filippo Bistaffa. Visgraphvar: A benchmark generator for assessing variability in graph analysis using large vision-language models. *IEEE Access*, 13: 21788–21810, 2025. doi: 10.1109/ACCESS.2025.3535837.

Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context. *arXiv preprint arXiv:2404.18532*, 2024.

Yuanfu Sun, Yuanhang Ren, Kang Li, Chuanhao Ji, Jiaxi Li, Jiajin Liu, Ninghao Liu, and Qiaoyu Tan. Graphverse: A comprehensive visual graph reasoning benchmark for multimodal large language models. *arXiv preprint arXiv:2608.06769*, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. In *International Conference on Learning Representations*, volume 2025, pp. 62624–62650, 2025a.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36:30840–30861, 2023.

Zhikai Wang, Jiashuo Sun, Wenqi Zhang, Zhiqiang Hu, Xin Li, Fan Wang, and Deli Zhao. Benchmarking multimodal mathematical reasoning with explicit visual dependency. *arXiv preprint arXiv:2504.18589*, 2025b.

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. *Advances in neural information processing systems*, 37:44–72, 2024.

Anyi Xu, B Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, CC Yu, et al. Deepseek-v4. 1-flash: Pushing the limits of kv cache compression. *arXiv preprint arXiv:2609.19969*, 2026a.

Mufan Xu, Kehai Chen, Xuefeng Bai, Muyun Yang, Tiejun Zhao, and Min Zhang. Structure pruning with llms for knowledge graph question answering. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–12, 2026b.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. *arXiv preprint arXiv:2406.12742*, 2024.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. Benchmarking and improving large vision-language models for fundamental visual graph understanding and reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30678–30701, July 2025.

Table 6: Examples of the eight end-to-end graph-theoretic reasoning tasks. For tasks admitting multiple valid solutions, one valid answer is shown. Long structured answers are abbreviated for readability.

ID	Task	Example question	Example answer
Partial graph views			
TS	Topological Sorting	Merge all views into a directed acyclic graph and return a valid topological ordering of every node.	[E13, E9, E63, E11, E4, E10, E3, E18, E8, E48, E43, E53]
MF	Maximum Flow	After merging all capacity-network views, what is the maximum flow from source `E4` to sink `E6`?	46
BM	Bipartite Matching	Given the displayed applicant and job partitions, return a maximum-cardinality matching in the merged graph.	[(E11, E2), (E33, E8), (E35, E3), (E43, E80), (E55, E38), (E58, E1)]
HP	Hamiltonian Path	Merge all views and return a path that visits every node in the resulting graph exactly once.	[E9, E11, E67, E63, E13, E5, E70, E61, E12, E1, E7, E52, E80, E32]
Temporal graph views			
STS	Stable Topological Sorting	Return a valid topological order at every time step while minimizing inversions among nodes shared by consecutive snapshots.	π_1: [E19, E8, E3, E48, E9, E11, E37, E1] π_2: [E54, E8, E3, E48, E11, E1, E9, E37]
MFC	Maximum-flow Change	Between which consecutive snapshots does the maximum flow from `E5` to `E6` increase the most?	[2, 3]
SBM	Stable Bipartite Matching	Return a maximum-cardinality matching at every time step while minimizing assignment changes for stable applicants.	M_1: [(E34, E12), ..., (E26, E49)] M_2: [(E34, E12), ..., (E68, E24)]
THP	Time-respecting Hamiltonian Path	Return a Hamiltonian path and the nondecreasing image index used by each path edge, using every time step.	Path: [E14, E43, E6, E57, E8, E21, E20, E9, E36, E34, E2, E12, E51, E26] Images: [1,1,1,1,1,1,1, 2,2,2,2,2,2]

A Details of Dataset Construction

This appendix describes the graph sources, view generation, question and answer construction, rendering, and validation procedures used in MVGraph. Table 6–9 illustrates the examples of each task. We also detail how the stage-oriented diagnostic tasks are constructed from the same visual-graph setting to probe perception, alignment, composition, and reasoning.

A.1 Construction Pipeline

An instance is built by generating or sampling a task-conditioned graph structure, constructing interrelated views, instantiating a question and its programmatically computed answer, applying task-specific checks, and rendering the accepted views as images. The views form an indexed sequence $\mathcal{I} = (I_1, \ldots, I_m)$, where $2 \leq m \leq 8$. The eight end-to-end graph-theoretic tasks and their structure-prepared reasoning conditions use two to four views; the remaining diagnostic tasks can use two to eight views. In particular, MFC uses three or four temporal snapshots. The order of the views is retained for temporal questions.

Each image I_i is rendered from a graph $G_i = (V_i, \mathcal{E}_i)$, with node set V_i and edge triples $\mathcal{E}_i \subseteq V_i \times \mathcal{R} \times V_i$. The same node identifier denotes the same entity across views, but node positions are determined independently for each image. Both node and edge sets may differ between views; shared coordinates are therefore neither assumed nor needed for cross-view identity matching. For perception, alignment, and composition, the model receives the graph images. In the perception protocol, it receives multiple images together and produces a separate textual node-and-edge representation for each view. For reasoning diagnostics, the model instead receives task-prepared textual graph structure or composition evidence, as described below.

Table 7: Examples of the ten Alignment tasks. For tasks admitting multiple valid solutions, one valid answer is shown. Long structured answers are abbreviated for readability.

ID	Task	Example question	Example answer
Partial graph views			
A-SN	Shared Nodes	Identify the node labels that are present in all graph views.	[E4, E61]
A-VSN	View-Specific Nodes	List the nodes that are exclusive to Graph 3.	[E30]
A-SE	Shared Edges	What edge triples are shared by every graph view?	[[E11, R1, E56]]
A-VSE	View-Specific Edges	What edge triples are unique to Graph 1 across the given graph views?	[[E1, R4, E61], [E4, R1, E71], [E15, R2, E61]]
Temporal graph views			
A-AN	Added Nodes	Identify the nodes newly appearing between t_2 and t_3.	[E1]
A-RN	Removed Nodes	Identify the nodes disappearing between t_1 and t_2.	[E37]
A-AE	Added Edges	Identify the edge triples newly appearing between t_3 and t_4.	[[E4, R2, E65], [E29, R3, E38], [E47, R1, E65]]
A-RE	Removed Edges	Which edge triples are present in t_3 but absent in t_4?	[[E28, R1, E61], [E59, R4, E71]]
A-PN	Persistent Nodes	List the nodes that remain present throughout the whole sequence.	[E31, E59]
A-PE	Persistent Edges	Which edge triples are present at every time step?	[[E20, R1, E71]]

Table 8: Examples of the eight Composition tasks. For tasks admitting multiple valid solutions, one valid answer is shown. Long structured answers are abbreviated for readability.

ID	Task	Example question	Example answer
Partial graph views			
C-UN	Union Node Count	How many unique node labels appear in the union of all graph views?	22
C-UE	Union Edge Count	Across all graph views, how many different edge triples are shown?	28
C-PA	Path Assembly	After merging all graph views, give one path from E28 to E21 as an ordered list of edge triples.	[[E28, R1, E77], [E77, R2, E58], [E58, R3, E13], [E13, R4, E50], [E50, R1, E78], [E78, R2, E21]]
C-CD	Component Decomposition	After merging all graph views, list every connected component with its nodes and edge triples.	component: nodes [E13,E19,E22,E27, E28,E38,E44]; edges [[E13,R1,E27], ...] component: nodes [E26,E45,E58,E78]; edges [[E26,R3,E45], ...]
C-MVS	Minimal View Set	Which graph views form the smallest sufficient set for connecting E5 to E76?	[1, 2]
C-EV	Essential Views	Which graph views are indispensable for making E74 reachable from E55?	[2, 3, 4]
Temporal graph views			
C-PP	Persistent Path	In the node-and-edge intersection graph, give one path from E23 to E70.	[[E23, R1, E79], [E79, R2, E16], [E16, R3, E71], [E71, R4, E70]]
C-TPC	Temporal Path Composition	Find adjacent times where E60 and E53 change from disconnected to connected. Report the two prior components and the later path.	t_1: subgraph1 [[E27,R2,E48], ...]; subgraph2 [[E20,R4,E76], ...]; t_2: path [[E60,R1,E75],..., [E76,R1,E53]]

A.2 Graph Sources and View Generation

Task-conditioned synthetic graphs. Synthetic generators construct candidate graphs around the structural properties needed by each task rather than sampling arbitrary graphs and searching retro-

Table 9: Examples of the ten reasoning tasks. For tasks admitting multiple valid solutions, one valid answer is shown. Long structured answers are abbreviated for readability.

ID	Task	Example question	Example answer
Partial graph views			
R-TS	Topological Sorting	Merge all views into a directed acyclic graph and return a valid topological ordering of every node.	[E45, E63, E72, E22, E31, E65, E47, E48, E7, E30, E44, E49, E24, E62]
R-MF	Maximum Flow	After merging all capacity-network views, what is the maximum flow from source `E37` to sink `E79`?	55
R-BM	Bipartite Matching	Given the applicant and job partitions in the graph facts, return a maximum-cardinality matching in the merged graph.	[(E24, E11), (E39, E3), (E5, E6), (E51, E17), (E62, E35), (E63, E47), (E66, E28), (E76, E38)]
R-BN	Bridge-Node Criticality	Which cross-view node(s) act as necessary bridges on the route from `E60` to `E47`?	[E28, E31, E75]
R-HP	Hamiltonian Path	Merge the graph facts into one undirected graph and return a path that visits every node exactly once.	[E10, E16, E59, E33, E5, E22, E72, E3, E27, E49, E28, E39, E50, E73, E18, E55, E30]
Temporal graph views			
R-STS	Stable Topological Sorting	Give one valid topological ordering for each time step, in image order. Among all valid sequences, minimize the total number of pairwise order inversions for nodes shared by consecutive time steps.	π_1: [E79, E24, ..., E5, E40, E45] π_2: [E66, E23, ..., E25, E40, E45] π_3: [E79, E40, ..., E24, E45, E59] π_4: [E40, E21, ..., E45, E79, E23]
R-MFC	Maximum-flow Change	The source is `E74` and the sink is `E54`. Between which consecutive graph images does the maximum-flow value increase the most? Return the two image numbers in chronological order.	[2, 3]
R-SBM	Stable Bipartite Matching	Return a maximum-cardinality matching for every time step. Among all such sequences, minimize how many stable applicants change their assigned job between consecutive images.	M_1: [(E39, E46), (E44, E54), ...] M_2: [(E39, E54), (E44, E62), ...] M_3: [(E39, E1), (E44, E18), ...] M_4: [(E39, E2), (E44, E18), ...]
R-THP	Time-respecting Hamiltonian Path	Find a time-respecting Hamiltonian path that visits every node exactly once. For each path edge, give the graph image in which it is used; image numbers must be nondecreasing and every time step must be used.	Path: [E47, E73, ..., E48, E65] Images: [1,1,1,1,2,2,...,3,4,4,4,4]
R-CE	Causal Edge Identification	Given the graph facts and composition evidence, which newly added edge triples are individually necessary for `E68` to reach `E52`?	[[E23, R3, E42], [E42, R4, E61]]

spectively for suitable questions. For partial views, the generators control shared and view-specific nodes and edges, allocate complementary path segments, and add distractor structures. For temporal views, they apply controlled node or edge changes to successive snapshots and select transitions or structures that satisfy the target question. These procedures retain the original local-view construction rules while organizing the resulting diagnostic tasks according to their primary target capability.

Alignment tasks control cross-view overlap or temporal change in nodes and complete edge triples. The partial-view tasks are shared nodes (A-SN), view-specific nodes (A-VSN), shared edges (A-SE), and view-specific edges (A-VSE). The temporal tasks are added nodes (A-AN), removed nodes (A-RN), added edges (A-AE), removed edges (A-RE), persistent nodes (A-PN), and persistent edges (A-PE). Relation identifiers may occur on multiple edges, so edge identity is determined by the complete (u, r, v) triple, not by r alone. Matching the same node across views relies on its shared identifier; the tasks do not introduce aliases or an additional entity-linking problem.

Composition tasks require organizing evidence distributed across views. For partial views, union node count (C-UN) and union edge count (C-UE) operate on the deduplicated union; path assembly (C-PA) requests a source–target path as an ordered sequence of edge triples, or `NONE` when no such path exists; and component decomposition (C-CD) requests every component of the union graph with its nodes and edges, including isolated nodes. Minimal view set (C-MVS) identifies a smallest subset of views that preserves source–target reachability, whereas essential views (C-EV) identifies views whose individual removal destroys it. Their generators distribute path evidence across views and introduce other structures so that the designated view contributions can be tested. For temporal

views, persistent path (C-PP) searches the intersection of the node and edge sets of all snapshots for a source–target path, returning `NONE` if absent. Temporal path composition (C-TPC) constructs a transition at which initially separate source and target components first become connected; its answer contains the two complete pre-transition component subgraphs and a post-transition connecting path.

The structure-prepared *reasoning* diagnostics reuse the eight graph-theoretic problems described below and additionally include bridge-node criticality (R-BN) and causal edge identification (R-CE). R-BN uses an already composed graph and its cross-view shared-node list to test which shared nodes disconnect a designated source and target when removed. R-CE uses ordered temporal structures and prepared composition evidence to identify newly added edges necessary for the queried reachability transition. The diagnostic preparation changes the information supplied to the model, not the underlying graph-theoretic question.

SmallPedia topologies. To complement controlled synthetic structures with naturally occurring connectivity patterns, we also draw local topologies from the static and temporal components of TKGL-SMALLPEDIA in TGB 2.0 (Gastinger et al., 2024) for applicable diagnostic instances. SmallPedia supplies topology only; its original images, entity names, and semantic question–answer pairs are not reused. We remove self-loops, treat sampled structures as undirected, and deterministically retain one relation when multiple facts connect the same unordered node pair. Entities and relations are anonymized as `E*` and `R*`, and the resulting graphs are rendered in the same style as synthetic instances. The sampled edges and anonymization mappings are retained in instance metadata for traceability. For partial views, sampled local static structures are partitioned into overlapping subgraphs while preserving their sampled connectivity. For temporal views, local structures are constructed from timestamped facts in their original order; a transition question identifies the queried pair of consecutive snapshots. Questions and answers are generated from these transformed anonymous graphs, not from the original entity semantics.

Graph-theoretic challenge generators. The eight end-to-end tasks are generated using randomized procedures specialized to four graph-theoretic problem families. For the partial-view tasks, a complete problem graph is generated and validated before its edges are distributed across two to four views. Edge allocation is balanced, and each view's node set is determined by its incident edges and additional nodes sampled from the complete graph. For the temporal tasks, successive graphs contain persistent and time-specific nodes, edges, or capacities. Candidates are filtered by exact algorithms and problem-specific non-degeneracy constraints.

Topological sorting. For TS, we generate a directed acyclic graph with 12–14 nodes and distribute its edges over two to four weakly connected local views. Each view contributes ordering constraints not fully implied by the other views, and the complete graph admits multiple valid topological orders while retaining a sufficiently long dependency chain. For STS, each snapshot contains seven stable nodes and one time-specific node. The answer gives a valid order for each snapshot while minimizing the total Kendall inversion distance between the relative orders of nodes shared by consecutive snapshots.

Maximum flow. For MF, a directed capacity network with designated source and sink is distributed across two to four partial views. Candidates are retained only when removing any view decreases the maximum flow of the composed network. For MFC, three or four capacity-network snapshots contain nine stable core nodes and one or two time-specific nodes. Changes to edges and capacities yield different source–sink maximum-flow values, and the answer identifies the consecutive pair with the largest flow increase.

Bipartite matching. For BM, a bipartite graph with 8–10 applicant nodes and 8–10 job nodes is distributed across two to four partial views. The generator introduces a controlled Hall deficiency and preserves replaceable matching edges, so the maximum-cardinality solution need not match every applicant or be unique. For SBM, each snapshot contains seven stable applicants, seven stable jobs, one time-specific applicant, and one time-specific job. Candidate edges vary across time; the answer attains maximum matching cardinality in every snapshot while minimizing the number of job changes by stable applicants between consecutive snapshots.

Hamiltonian paths. For HP, connected undirected graphs of 14–17 nodes are filtered using exact memoized Hamiltonian-path search before their edges are distributed across two to four views. Candidates without a Hamiltonian path or with trivial degree patterns are rejected. For THP, two to

four temporal views have different node and edge sets, with 7–11 nodes in each view. An accepted sequence admits a Hamiltonian path whose edge-image indices are nondecreasing and collectively use every time step. The answer contains both the node sequence and the image index assigned to each path edge.

A.3 Graph Representation and Rendering

The partial-view and temporal diagnostic graphs derived from the original foundational generators, including their SmallPedia-derived counterparts, are undirected. Among the graph-theoretic tasks, topological sorting and maximum flow use directed graphs, whereas bipartite matching and Hamiltonian paths use undirected graphs. Nodes have semantically neutral `E*` identifiers and reusable relation types have `R*` identifiers; maximum-flow edges additionally display positive numerical capacities. These labels prevent domain-specific lexical semantics from providing the answer.

Each rendered view contains 7–20 nodes, a bound applied to individual images rather than to the union of all views. Layouts are computed independently from local graph structures, so recurring nodes can appear at different positions. Node shapes, fonts, edge styles, relation labels, arrowheads where applicable, and image titles follow a unified rendering style. Graph-theoretic layouts are additionally checked for node spacing and, for maximum-flow networks, overlap involving capacity labels. Layouts that fail the applicable visual checks are regenerated.

A.4 Question and Answer Generation

Task-specific templates instantiate questions after the graph views have been finalized. The output format depends on the task: answers can be counts, node or edge-triple lists, view indices, component subgraphs, paths, topological orders, matchings, or task-specific temporal outputs. The perception protocol instead requests a separate textual representation of the nodes and edge triples in every input image. All reference answers are computed from the underlying graph data rather than from model-generated annotations.

For alignment, answers are obtained through node-set or edge-triple intersection and difference across partial views or designated temporal snapshots. For composition, the construction computes deduplicated unions, components, path witnesses, persistent structures, or sufficient and indispensable view subsets as specified by the task. For end-to-end graph-theoretic tasks, exact graph algorithms compute reference orders, flow values, matchings, and Hamiltonian paths, including temporal stability or edge-time constraints where applicable. R-BN and R-CE use the prepared structural evidence to formulate node-removal and newly added-edge necessity queries, respectively.

The reasoning-diagnostic input is prepared from the same underlying structures. For the four partial-view graph-theoretic tasks, the graph is already composed into a textual representation; for temporal tasks, the snapshot sequence and its order are preserved. R-BN additionally supplies the composed graph and the list of shared candidate nodes, whereas R-CE supplies the temporal structures and composition evidence needed to locate the relevant transition. Thus these conditions target graph-theoretic and counterfactual inference after the task-relevant structure has been made explicit.

Several graph-theoretic questions admit multiple correct outputs, so the stored answer is a reference rather than the sole acceptable response. The topological-order verifier checks node coverage, uniqueness, and precedence constraints. The matching verifier checks partition membership, edge validity, one-to-one assignments, and maximum cardinality. The Hamiltonian-path verifier checks node coverage, uniqueness, adjacency, and, for THP, nondecreasing and complete coverage of the required edge-image indices. For STS and SBM, the verifier additionally checks the respective minimum inversion or reassignment cost across snapshots.

A.5 Evidence Necessity and Validation

For tasks intended to test cross-view dependence, a view is treated as required when removing it changes the programmatically established answer or leaves insufficient information to establish that answer under the task semantics. Where available, required-view indices are retained in the metadata and are not supplied to the evaluated model. When a question remains well-defined after removing a view, leave-one-out answer recomputation compares the reduced-view and full-view answers. For

Table 10: Per-task statistics for the end-to-end graph-theoretic challenges. Average node and edge counts are computed after deduplication across all views of an instance.

ID	Task	Instances	Views	Avg. views	Avg. nodes	Avg. edges
Partial graph views						
TS	Topological Sorting	1,000	2–4	2.98	13.01	36.23
MF	Maximum Flow	1,000	2–4	2.98	12.00	25.54
BM	Bipartite Matching	1,000	2–4	3.02	18.00	34.28
HP	Hamiltonian Path	1,000	2–4	3.03	15.51	40.91
Temporal graph views						
STS	Stable Topological Sorting	1,000	2–4	3.03	10.03	32.01
MFC	Maximum-flow Change	1,000	3–4	3.49	14.19	24.47
SBM	Stable Bipartite Matching	1,000	2–4	2.98	19.95	50.79
THP	Time-respecting Hamiltonian Path	1,000	2–4	2.97	15.49	28.02
Total		**8,000**	**2–4**	**3.06**	**14.77**	**34.03**

Table 11: Per-task statistics for the alignment diagnostics.

ID	Task	Instances	Views	Avg. views	Avg. nodes	Avg. edges
Partial graph views						
A-SN	Shared Nodes	1,000	2–8	4.78	33.10	47.51
A-VSN	View-Specific Nodes	1,000	2–8	4.76	32.91	40.19
A-SE	Shared Edges	1,000	2–8	4.72	33.90	44.91
A-VSE	View-Specific Edges	1,000	2–8	4.72	32.71	41.10
Temporal graph views						
A-AN	Added Nodes	1,000	3–8	4.98	10.00	13.14
A-RN	Removed Nodes	1,000	3–8	4.99	9.99	13.28
A-AE	Added Edges	1,000	3–8	4.95	13.43	10.58
A-RE	Removed Edges	1,000	3–8	5.00	13.45	10.57
A-PN	Persistent Nodes	1,000	3–8	4.99	34.63	42.41
A-PE	Persistent Edges	1,000	3–8	5.00	26.56	21.21
Total		**10,000**	**2–8**	**4.89**	**24.07**	**28.49**

tasks such as intersections, persistent properties, and negative reachability, unchanged recomputed values alone do not establish that a view was redundant, because the omitted view may be needed to rule out alternatives. The perception protocol is different: it tests reconstruction of each supplied view and does not impose a cross-view evidence-necessity criterion on its output.

Task-specific construction checks also enforce the intended structural dependencies. For example, TS requires independent ordering constraints across views, MF requires every view to contribute to the maximum-flow value, and temporal tasks retain the snapshots needed to establish their queried transition or persistent property. Diagnostic tasks use their corresponding overlap, path-distribution, temporal-change, and evidence-selection constraints rather than a single universal leave-one-out test.

Across tasks, structural checks require edge endpoints to belong to their view's node set, exclude disallowed self-loops and duplicate edges, and enforce the prescribed per-view node bounds. Task-specific checks verify properties such as acyclicity, bipartite structure, transition direction, path existence, and solution feasibility where relevant. Stored reference answers are checked against the finalized graph structures, and tasks with multiple valid solutions use the semantic verifiers described above. Non-degeneracy rules are applied by task, rather than rejecting empty outputs globally; in particular, C-PA and C-PP can legitimately return `NONE`. Finally, views that fail applicable readability checks, including node-spacing and label-placement constraints, are regenerated.

Table 12: Per-task statistics for the composition diagnostics.

ID	Task	Instances	Views	Avg. views	Avg. nodes	Avg. edges
Partial graph views						
C-UN	Union Node Count	1,000	2–8	4.71	32.51	39.72
C-UE	Union Edge Count	1,000	2–8	4.70	32.70	39.67
C-PA	Path Assembly	1,000	2–8	4.75	29.48	24.91
C-CD	Component Decomposition	1,000	2–8	4.75	12.73	10.22
C-MVS	Minimal View Set	1,000	2–8	4.79	31.41	26.45
C-EV	Essential Views	1,000	2–8	4.71	30.90	26.02
Temporal graph views						
C-PP	Persistent Path	1,000	3–8	5.00	18.91	17.91
C-TPC	Temporal Path Composition	1,000	3–8	5.04	14.02	12.48
Total		**8,000**	**2–8**	**4.81**	**25.33**	**24.67**

Table 13: Per-task statistics for the structure-prepared reasoning diagnostics. The eight graph-theoretic tasks reuse the underlying instances in Table 10, but supply structural information rather than requiring visual graph reconstruction.

ID	Task	Instances	Views	Avg. views	Avg. nodes	Avg. edges
Partial graph views						
R-TS	Topological Sorting	1,000	2–4	2.98	13.01	36.23
R-MF	Maximum Flow	1,000	2–4	2.98	12.00	25.54
R-BM	Bipartite Matching	1,000	2–4	3.02	18.00	34.28
R-HP	Hamiltonian Path	1,000	2–4	3.03	15.51	40.91
R-BN	Bridge-Node Criticality	1,000	2–8	4.78	29.69	25.08
Temporal graph views						
R-STS	Stable Topological Sorting	1,000	2–4	3.03	10.03	32.01
R-MFC	Maximum-flow Change	1,000	3–4	3.49	14.19	24.47
R-SBM	Stable Bipartite Matching	1,000	2–4	2.98	19.95	50.79
R-THP	Time-respecting Hamiltonian Path	1,000	2–4	2.97	15.49	28.02
R-CE	Causal Edge Identification	1,000	3–8	5.05	14.02	12.51
Total		**10,000**	**2–8**	**3.43**	**16.19**	**30.98**

A.6 Dataset Organization and Scale

Each instance record stores the rendered graph views, its question and gold answer, per-view graph structures in JSON, and instance-level metadata. The metadata identifies the task, image paths, and answer format, and records construction and validation information where applicable. For SmallPedia-derived instances, the source edges and anonymization mappings are also retained. Models do not receive the underlying graph JSON or construction metadata in image-based evaluation. In the structure-prepared reasoning condition, textualized structural evidence is instead supplied to probe downstream reasoning.

MVGraph comprises 37 evaluation tasks and 72,000 task-instance records across five data collections. The eight end-to-end graph-theoretic tasks contribute 8,000 records. The diagnostic collections comprise 36,000 perception, 10,000 alignment, 8,000 composition, and 10,000 reasoning records. Perception uses a single evaluation objective, reconstructing the nodes and edge triples of each supplied graph view, across 36,000 multi-view instances with 2–8 views and an average of 4.46 views. These instances contain, on average, 21.47 distinct nodes and 26.93 distinct edge triples across their views. Tables 10–13 provide per-task statistics for the remaining four groups.

B Inference Prompts and Instructions

This appendix summarizes the prompts and instructions provided to models during inference. Each input combined shared stage-level instructions with sample-specific graph context, a task question, and an output requirement. Bracketed placeholders indicate content instantiated for each sample; they were not shown literally to the model.

B.1 End-to-End Graph-Theoretic Reasoning

The end-to-end tasks received the ordered graph images directly. Each sample used the following structure:

End-to-end prompt structure

[Scenario statement]

[Task-specific question]

Return only a valid JSON object of the form *[task-specific answer schema]*. Do not include explanation or extra fields.

The scenario statement specified whether the images were partial views or temporal snapshots, whether edges were directed, and whether edge labels represented relation types or numeric capacities. Identical `E*` labels denoted the same node across images. For non-capacity tasks, edges were triples [head node, relation type, tail node], where `R*` was a reusable relation type. For capacity tasks, the displayed number on each directed edge was its capacity; relation labels were not shown.

End-to-end task questions

TS (RH1). Merge all views into one directed acyclic graph. Give a valid topological ordering of every node in the merged graph. The answer is a JSON list of node labels.

STS (RH2). Give one valid topological ordering for each time step, in image order. Among all valid sequences, minimize the total number of pairwise order inversions for nodes shared by consecutive time steps. The answer is a JSON list of node-label lists.

MF (RH3). Merge all views into one capacity network. The source is *[source]* and the sink is *[sink]*. What is the maximum-flow value from source to sink? The answer is a JSON integer.

MFC (RH4). The source is *[source]* and the sink is *[sink]*. Between which consecutive graph images does the maximum-flow value increase the most? Return the two graph image numbers in chronological order. The answer is a JSON list of two image numbers.

BM (RH5). The applicant partition is *[applicants]*, and the job partition is *[jobs]*. Find a maximum-cardinality assignment in the merged graph. Return applicant-job pairs. The answer is a JSON list of pairs.

SBM (RH6). The applicant partitions in image order are *[applicants by time]*, and the job partitions are *[jobs by time]*. Return a maximum-cardinality matching for every time step. Among all such sequences, minimize how many stable applicants change their assigned job between consecutive images. The answer is a JSON list of matchings.

HP (RH7). Merge all views into one undirected graph. Give a Hamiltonian path that visits every node in the merged graph exactly once. The answer is a JSON list of node labels.

THP (RH8). Find a time-respecting Hamiltonian path that visits every node exactly once. For each path edge, give the graph image in which that edge is used. Image numbers must be nondecreasing, and every time step must be used. The answer is a JSON object containing `path` and `images` lists.

B.2 Perception

Models received the graph images in order. The common image-parsing instruction was:

Perception prompt

You are a graph image parsing assistant.

[Graph scenario description]

Inspect all graph images and reconstruct every graph shown. Do not infer relation meanings or facts that are not visible in the images.

Return exactly one JSON object containing a `graphs` list. For each graph, preserve its `graph_id` and output its `triples`. Extract every node, including isolated nodes; represent an isolated node as `["E80", "", ""]`. Preserve node and edge labels exactly. Do not include explanations, Markdown, or additional keys.

The undirected variant emitted each edge once. The directed variant preserved arrow direction without adding reverse edges. For capacity networks, the image-parsing output used the visible numeric capacity rather than inventing an unshown relation label.

B.3 Alignment and Composition

These image-based tasks shared a scenario preamble but had no separate stage-specific system prompt. For partial views, the preamble was:

Partial-view scenario preamble

The images show partial views of related undirected graphs. Same `E*` labels indicate the same node across views, and edges are triples [head node, relation type, tail node] where `R*` is the reusable relation type.

For temporal views, it was:

Temporal-view scenario preamble

The images show an undirected graph at different time steps. Same `E*` labels indicate the same node across time, and edges are triples [head node, relation type, tail node] where `R*` is the reusable relation type.

Each preamble was followed by the task-specific question and its JSON output instruction. Alignment questions requested shared, view-specific, added, removed, or persistent nodes and edge triples. Composition questions requested union statistics, assembled paths, component subgraphs, necessary views, or temporal structural evidence. The required JSON schema depended on the task; it was not uniformly a node list.

B.4 Reasoning

Models received prepared graph facts rather than images. The common instruction began:

Reasoning prompt

You are a graph reasoning assistant.

Answer the question using only the graph facts and question provided below. Do not use images, outside knowledge, or unstated assumptions.

Graph text construction: *[Scenario, CSV columns, edge semantics, and isolated-node encoding.]* Treat node labels and relation labels as exact strings. Relation labels such as `R1`, `R2`, `R3`, and `R4` have no implicit meaning.

Graph facts: *[Graph CSV]*

Question: *[Task-specific question and JSON output instruction]*

Table 14: Performance of LVLMs on the 8 end-to-end graph-theoretic reasoning tasks of MV-GRAPH, where † denotes evaluation via API called. The best results are **bolded**.

Model	TS	STS	MF	MFC	BM	SBM	HP	THP	Avg.
No-Thinking									
Gemma-4-12B-it	**0.0**	**0.0**	0.0	40.9	0.0	**0.1**	**4.0**	0.0	5.6
Qwen3.5-9B	**0.0**	**0.0**	0.0	41.5	0.0	0.0	0.2	0.0	5.2
Ovis2.5-9B	**0.0**	**0.0**	0.0	41.7	0.0	0.0	0.3	0.0	5.2
Qwen3-VL-8B	**0.0**	**0.0**	0.0	42.3	0.0	0.0	1.0	0.0	5.4
GLM-4.6V-Flash	**0.0**	**0.0**	0.0	41.7	0.0	0.0	0.0	0.0	5.2
Molmo2-8B	**0.0**	**0.0**	**2.8**	42.5	0.0	0.0	0.0	0.0	**5.7**
MiniCPM-V-4.5	**0.0**	**0.0**	1.0	31.5	0.0	0.0	0.2	0.0	4.1
Pixtral-12B	**0.0**	**0.0**	0.0	15.4	0.0	0.0	0.0	0.0	1.9
InternVL3.5-8B	**0.0**	**0.0**	0.0	**42.7**	**0.1**	0.0	0.0	0.0	5.4
LLaVA-OneVision-2-8B	**0.0**	**0.0**	0.0	42.0	0.0	0.0	0.2	**0.1**	5.3
Phi-4-reasoning-vision-15B	**0.0**	**0.0**	0.1	42.6	0.0	0.0	0.1	0.0	5.4
Llama-3.1-Nemotron-Nano-VL-8B	**0.0**	**0.0**	0.0	40.6	0.0	0.0	0.0	0.0	5.1
DeepSeek-Flash†	**0.0**	**0.0**	**0.0**	**45.0**	**2.0**	**0.0**	**2.0**	**2.0**	**6.4**
Qwen3.8-Flash†	**0.0**	**0.0**	**0.0**	17.0	**2.0**	**0.0**	**2.0**	0.0	2.6
Thinking									
Qwen3.5-9B	**0.1**	0.0	0.3	30.2	36.9	0.2	40.0	0.4	13.5
Gemma-4-12B-it	0.0	**0.1**	**5.5**	45.3	**60.6**	**0.4**	**43.1**	**8.7**	**20.5**
GLM-4.6V-Flash	0.0	0.0	0.7	**46.5**	7.2	0.0	4.7	0.0	7.4
Ovis2.5-9B	0.0	0.0	0.2	41.3	18.5	0.0	2.3	0.2	7.8
Qwen3.8-Flash†	**35.0**	8.0	**54.0**	**59.0**	**97.0**	8.0	77.0	65.0	50.4
GLM-5.3-Flash†	33.0	**12.0**	41.0	**59.0**	95.0	**16.0**	**84.0**	**79.0**	**52.4**
DeepSeek-Flash†	0.0	0.0	15.0	5.0	62.0	0.0	50.0	7.0	17.4

For partial-view reasoning tasks, the supplied facts contained the merged graph. Temporal tasks retained separate snapshots in time order. The CSV description varied with edge direction and capacity; the causal-edge task additionally supplied Composition evidence.

C END-TO-END GRAPH-THEORETIC PERFORMANCE

C.1 MAIN RESULTS

Table 14 reports the complete results across inference settings. Without thinking, all 14 evaluated models achieve low average accuracy, ranging from 1.9% to 6.4%. Topological sorting (TS) and stable topological sorting (STS) receive no correct answers, while performance on most other tasks is also near zero. Maximum-flow change (MFC) is the main exception, with substantially higher scores for many models. These scores warrant caution: MFC asks models to select a pair of consecutive snapshots from a sequence of three or four, rather than to output or justify the maximum-flow values themselves. Its smaller answer space means that correct selections do not necessarily demonstrate reliable flow computation, and MFC contributes disproportionately to the no-thinking averages.

Thinking improves the average accuracy of every model evaluated in both settings, but the magnitude of the improvement varies considerably. Qwen3.8-Flash rises from 2.6% to 50.4%, and Gemma-4-12B-it rises from 5.6% to 20.5%, whereas the gains for GLM-4.6V-Flash and Ovis2.5-9B are much smaller. Among the thinking models, GLM-5.3-Flash and Qwen3.8-Flash perform best overall, followed at a substantial distance by the strongest locally evaluated model, Gemma-4-12B-it. DeepSeek-Flash performs below Gemma-4-12B-it despite being API-accessed, showing that the performance differences cannot be characterized simply by deployment category. These comparisons are descriptive rather than size-controlled, and GLM-5.3-Flash has no corresponding no-thinking result in this evaluation.

Performance also depends strongly on the graph-theoretic problem. The two leading models achieve 95.0–97.0% on bipartite matching (BM), but only 8.0–16.0% on stable bipartite matching (SBM). Their TS scores likewise exceed those on STS. The temporal gap is not uniform, however: both mod-

els retain comparatively strong performance on time-respecting Hamiltonian path (THP). Moreover, each static–temporal task pair differs in construction, output requirements, or both, so these score differences should not be interpreted as an isolated effect of temporal reasoning. Taken together, the full results show that additional test-time reasoning can yield large gains for some models and tasks, while leaving major graph-theoretic challenges unresolved. They motivate the stage-oriented diagnostics used to examine which capabilities limit end-to-end performance.