Multimodal LLMs Struggle with Basic Visual Network Analysis: A VNA Benchmark

Evan M. Williams(✉) and Kathleen M. Carley

Carnegie Mellon University, Pittsburgh, PA 15213, USA
{emwillia,carley}@andrew.cmu.edu

Abstract. We evaluate the zero-shot ability of GPT-4 and LLaVa to perform simple Visual Network Analysis (VNA) tasks on small-scale graphs. We evaluate the Vision Language Models (VLMs) on 5 tasks related to three foundational network science concepts: identifying nodes of maximal degree on a rendered graph, identifying whether signed triads are balanced or unbalanced, and counting components. The tasks are structured to be easy for a human who understands the underlying graph theoretic concepts, and can all be solved by counting the appropriate elements in graphs. We find that while GPT-4 consistently outperforms LLaVa, both models struggle with every visual network analysis task we propose. We publicly release the first benchmark for the evaluation of VLMs on foundational VNA tasks.

Keywords: Vision Language Models · Graphs · Benchmark

1 Introduction

Large Language Models (LLMs) and Large Vision Language Models (VLMs) are transforming the ways people work and research. There has recently been a surge of interest in understanding how LLMs can be used in network analytic workflows and evaluating their performance on network-level tasks [5]. As myriad Generative AI tools are introduced to assist practitioners with every aspect of data analytics pipelines, understanding the limitations of these tools is important. Network Analysis is used in almost every domain imaginable, and network visualization is often a core component of network analytics pipelines. Misinterpreting a graph can lead to wildly incorrect conclusions about the underlying data, which could lead practitioners to act in sub-optimal ways.[1] We introduce the broad task of zero-shot Visual Network Analysis (VNA), which we define as deriving graph theoretic concepts from network visualizations without relying on human-annotated examples. We introduce a new VNA benchmark and evaluate the performance of GPT-4 [1] and LLaVa1.5-3b [11] on 5 tasks based on 3 foundational network science concepts.

[1] At the time this was first submitted to ArXiv, there were not yet any public visual network analysis tasks. However, two datasets have since been released; we note that none of the tasks introduced in this work were considered in the concurrent works or benchmarks.

R. Thomson et al. (Eds.): SBP-BRiMS 2024, LNCS 14972, pp. 15–24, 2024.
https://doi.org/10.1007/978-3-031-72241-7_2

We introduce 5 VNA tasks based on three simple and important graph theory concepts: 1) degree, 2) structural balance, and 3) components. We create two tasks related to degree. Given a visualized graph, we ask the multimodal LLM to 1a) identify the maximum degree of the graph, i.e., the largest degree of any node, and 1b) to return the node IDs of all nodes with the maximum degree. We create one task based on structural balance: 2a) given an image of a triad with edges colored to denote a relation type, we ask the multimodal LLM to assess whether the triad is balanced or imbalanced. Finally, given an image of graphs with multiple components, 3a) we ask the LLM to count the number of components in the graph, and 3b) we ask the LLM to count the number of isolates.

Each of these tasks is related to an important graph theoretic concept, but each task is also connected in how it can be solved. Every task we create can be solved by counting the correct element within each graph. For the degree-level tasks, the multimodal LLM needs to count edges incident to the nodes that appear to have the largest number of edges. In the structural balance tasks, the answer can be deduced by either counting the number of positive or negative edges in the triad. The component tasks are very explicitly asking the multimodal LLM to count the number of components and the number of isolates. Consequently, these tasks, as we've constructed them, are highly related to zero-shot object counting [14]. For each of these tasks, we generate synthetic, high-resolution, graph visualizations while prioritizing human readability. We publish the all of the data generated for this project[2]

We find that GPT-4 and LLaVa struggle on all 5 tasks we propose. Across all experiments, the highest accuracy was achieved by GPT-4 on the isolate counting task, where it correctly identified the number of isolates in 67 of 100 graph visualizations. Predicting whether or not triads were structurally balanced was surprisingly one of the most challenging tasks for both LVMs, despite the simplicity of the task. The best performing model—GPT-4—achieved an overall accuracy of 0.51 on the task, on par with random guessing. More work is needed to understand and improve zero-shot LVM performance on visual graph analysis tasks.

2 Related Works

Recent work has explored potential Large Language Model (LLM) applications within social networks [15], and in generating features and predictions for machine learning applications on graphs [4]. Recent work has also explored how best to encode graphs as text for LLM analysis [5]. However, each of these works look solely at LLMs and do not consider VLM usage. Two days prior to the release of this publication on ArXiv, the VisionGraph Benchmark was released [10]. The visiongraph benchmark contains complementary tasks and evaluates

[2] https://figshare.com/articles/dataset/Multimodal_LLMs_Struggle_with_Basic_Visual_Network_Analysis_a_Visual_Network_Analysis_Benchmark/25938448.

VLM performance on many relatively-complex graph tasks, including cycle identification, identifying shortest paths, and identifying maximum flow [10]. Wei et al. concurrently introduced a GITQA (Graph-Image-Text Question Answering) Dataset and an end-to-end framework for general graph reasoning [13].

There has also been a recent surge in interest in zero-shot evaluation capabilities Large Vision Language Model (VLM) evaluations. These evaluations have been applied to a wide range of computer vision tasks [17]. [14] propose the task of zero-shot object counting, which they define as counting the number of instances of a specific class in the input image given only the class name. Llava has been evaluated on various zero-shot tasks, and has performed well on differentiating animals, counting animals, identifying written digits, and identifying pox [9]. Previous work has found that VLMs perform worse than specialized models at object counting, and seemed to perform better at counting cars than counting trees or animals [16].

3 Methods

In all tasks, each graph is independently sent to GPT-4 along with its accompanying text prompt using OpenAI's API. We also feed each graph, independently, to LLaVa, which we run on a local cluster. All LLaVa prompts were modified to include the suggested prompt format on which LLaVa was trained: "USER: <image> \n< prompt >\nASSISTANT:".

3.1 Maximum Degree Tasks

Degree centrality is likely the most widely-used graph centrality statistic. For an undirected graph G, the degree of node i is simply the number of edges incident to node i. For a binary, symmetric, and undirected adjacency matrix A, degree is simply defined as $\sum_i^N x_i$. In social networks, degree centrality often corresponds to popularity or engagement, e.g., in a friendship network, a node with a high degree has many friends.

We consider two different prompts for each graph. In the first prompt, we ask the question using Graph Theory terminology—we call this our "formal prompt". In the second question, we attempt to ask the question in a more human way. We state that the graph is a first-grade friendship network and ask the same questions using the terminology like popular students and total number of friends—we call this the "human prompt". We consider two prompts for each graph. The first asks for the largest degree centrality in the graph and the list of all nodes with that centrality. The second prompt states that we are observing a first-grade friendship network and asks who has the most friends. We exclude exact prompts from this paper due to space constraints, but we make all of prompts publicly available online[3]. LLaVa would return blank fields if given the same formatting stipulations as GPT-4, so those were dropped in

[3] https://github.com/EvanUp/VNA_Benchmark/blob/main/prompts/prompts.csv.

the LLaVa prompts. In the cases where LLaVa made contradictory statements or inaccurate statements, e.g., “A has a degree centrality of 10, while B has a degree centrality of 9. The largest degree centrality is B, which is 9”, we selected the most logically coherent option—in this case, (A, 10). In cases where LLaVa or GPT-4 did not return a numeric maximum degree or node IDs, we impute a maximum degree of 0 and assign the response an empty set of node IDs.

3.2 Structural Balance Task

Balance Theory, which dates back to Fritz Heider’s 1946 studies of individual cognition and perception of social situations, is a core concept in social network analysis [8]. Heider’s original formalization of *cognitive balance* was generalized in the 1950s to *structural balance*, which focuses on a group of people rather than an individual [3,7]. Given a undirected signed graph $\mathcal{G}^{+}_{-}$, where each edge can assume a value of “like” (+) or “dislike” (−), structural balance occurs when i likes j ($i \overset{+}{\leftrightarrow} j$) and i and j agree in their evaluation of k, i.e., $(i \overset{+}{\leftrightarrow} k \wedge j \overset{+}{\leftrightarrow} k) \vee (i \overset{-}{\leftrightarrow} k \wedge j \overset{-}{\leftrightarrow} k)$. Concretely, for signed, undirected, triads, this means that any triad with a total of 1 or 3 (+) edges is balanced. Conversely, triads with 0 or 2 (+) edges are considered unbalanced. Consequently, of the 8 possible signed triads, 4 are structurally balanced and 4 are unbalanced. For more information on structural balance we refer the reader to Chap. 6 of [12]. We considered two different prompts, one without a definition of structural balance and one containing a simple definition of structural balance.

3.3 Component Tasks

A component is a connected graph or subgraph that is not a part of a larger connected subgraph. In undirected graphs, the number of components can be defined as the minimum number of random walkers that it would take to reach every node on a graph. Nodes of degree zero are a special type of component commonly referred to as isolates. We considered two prompts for this task—one without definitions and one with definitions. Again, LLaVa was not generating responses with the formatting requirements included in the prompt, so its prompt was truncated. LLaVa’s return format with this prompt was largely consistent and answers could be reliably extracted with a regex. Of the 400 LVM calls for this task, there were 4 instances, all from GPT-4, where no numbers were returned. These instances were imputed with zeros.

4 Data

For each experiment, we programatically generate graphs using the python libraries NetworkX and netgraph. For the degree tasks, graphs were generated with Kamada Kawai layouts, as we qualitatively found these to be the most human-readable for visual node degree tasks. We bolded font weights in the degree and structural balance tasks for the same reason. All graphs are exported as high-quality 300 DPI .png files.

4.1 Maximum Degree Graph Generation

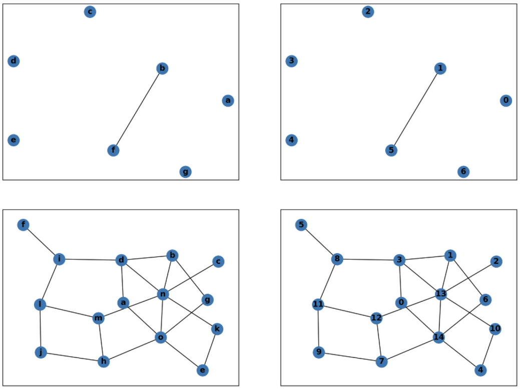

Fig. 1. Degree Task Graph Examples with letter (left) and numeric (right) node IDs.

We provide the LLMs with two degree-related tasks. First, we ask the LLM to find the maximum degree of the graph. Second, we ask the LLM to return the node IDs of all nodes with the maximum degree. In large graphs, this can be a challenging, or even infeasible, task for humans. Consequently, we consider only relatively-sparse graphs ranging from 1 to 20 nodes. We generate 20 Erdos-Renyi graphs, each with the parameter p set to 0.2. As LVMs have previously been found to be prone to typographic attacks [6], we considered that numeric node IDs could impact the ability of LLMs to return numbers. Consequently, we generate two identical versions of each graph: one with numeric node IDs and one with alphabetical node IDs (see Fig. 1).

4.2 Structural Balance Graph Generation

For each of the 8 possible signed triads, we generate 10 graphs, each with random layouts for a total of 80 triad graphs. In each graph, “like” edge relations are colored blue and “dislike” edge relations are colored red. As node IDs are unimportant for this task, we arbitrarily chose to use letter node IDs. We further group types of triads into four “classes” corresponding to the number of “like” (blue) edge relations. This results in two balanced groups—3b and 1b—containing 3 and 1 “like” relationship respectively and two unbalanced groups (0b and 2b in Fig. 2). In Fig. 2, we provide examples of 4 individual triads sampled from each grouping. We note that we allow position to vary across triad images, as it should be irrelevant to this task.

4.3 Component Graph Generation

For each component graph, we independently generate 4 Erdos-Renyi graphs, with $p = 0.3$ and where the number of nodes for each of the 4 graphs are

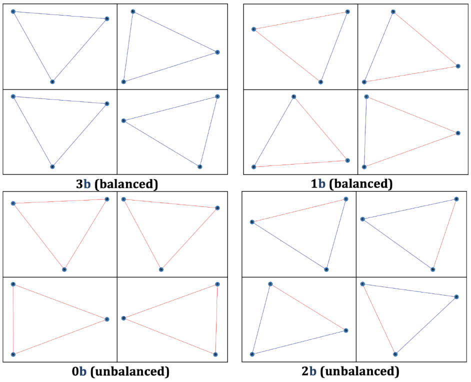

Fig. 2. Triadic Balance Examples. Top row contains a sample of balanced triads, bottom row contains a sample of unbalanced triads. ‘b’ denotes the number of like (blue) relationships in each group. (Color figure online)

randomly drawn integers between 0 and 30 inclusive. We then take the disjoint union between these four graphs, record the number of components and isolates in each graph, and visualize the result. Optimizing for human readability, we elected to visualize these graphs using the netgraph library, as it is optimized to support visual layouts containing multiple components [2]. We chose a small-world layout for each component again for human readability. Component counts across all graphs ranged from 2 to 11 and isolate counts ranged from 0 to 6. We provide four example component graphs in Fig. 3.

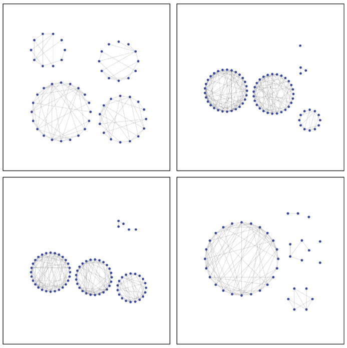

Fig. 3. Components Example Graphs. Read from left to right and top to bottom, these graphs contain 4, 5, 6, and 7 components respectively. The graphs contain 0, 1, 2, and 3 isolates.

5 Results

5.1 Maximum Degree Task Results

We evaluate GPT-4 and LLaVa at both 1) identifying the maximum node degree in a graph and 2) identifying node IDs that have the maximal degree. Despite being able to define degree centrality, LLaVa performed extremely poorly on both tasks. On the "formal" prompt, where we instruct LLaVa to identify nodes the maximum degree and identify all nodes that have the maximal degree, LLaVa failed to correctly identify any maximum degrees. On the human prompts, it exhibited bizarre behavior; it frequently made up its own graphs in its responses and then (often incorrectly) answered the question based on the graph it generated. In both scenarios, its predictions often deviated substantially from ground truth, even with easy examples, as can be seen in the relatively high MSE scores in Table 1. In the top row of graphs in Fig. 1, which very clearly has a maximum degree of one shared by two nodes, LLaVa's 4 runs identified maximum degrees of 3, 4, 10, and 11 respectively.

GPT-4 performed better than LLaVa on all metrics, but still struggled despite the simplicity of the task. A human-centric prompt with Letter IDs performed the best, yielded the best accuracy in identifying maximum degree, but did not have the best Mean Squared Error (MSE) nor the best Mean Jaccard Similarity. The human-centric prompt with letter IDs was the only run where GPT-4 correctly predicted the maximum degree of the graph in the top left corner of Fig. 1. In all other runs, GPT-4 incorrectly identified the maximum degree of the top row graphs as 2.

Table 1. Results of GPT-4 and LLava on Identifying the maximum degree (Accuracy, MSE), and at identifying the set of nodes IDs that have the maximum degree in the graph (mean Jaccard similarity over all graphs)

			Max Degree		Degree IDs
			Accuracy	MSE	Mean Jaccard Similarity
GPT-4	Formal Prompt	Numeric IDs	0.45	**17**	0.491
		Letter IDs	0.35	24	0.53
	Human Prompt	Numeric IDs	0.4	23	**0.54**
		Letter IDs	**0.533**	25	0.533
LLaVa	Formal Prompt	Numeric IDs	0	693	0.075
		Letter IDs	0	328	0.195
	Human Prompt	Numeric IDs	0.15	760	0.065
		Letter IDs	0.1	572	0.054

5.2 Structural Balance Task Results

GPT-4 generally performed well on the cases with 3 (+) edges (b1 in Fig. 2) and 0 (+) edges (b0), achieving accuracies greater than or equal to 0.7 for balanced and unbalanced triads in these categories. However, it did surprisingly poorly across cases with 1 or 2 (+) edges. Balanced triads with 1 edge (b1) with and without a definition in the prompt received accuracies of 0.2 and 0.5, respectively. Unbalanced triads with 2 (+) edges (2b) received accuracies of 0.467 and 0.367 respectively. We note that GPT-4's reasoning was often inconsistent or faulty, even when a clear definition of structural balance was provided in the prompt. For example, on one of the (b0) categories that it incorrectly classified as balanced, GPT-4 returned the justification: "This triad is balanced as all three edges depict "dislike" relationships (even number of "dislike", odd number of "like")".

When provided with a clear definition of what constitutes structural balance, LLaVa predicted that every triad was unbalanced. Without a definition, LLaVa still largely overwhelmingly predicting that triads were unbalanced. In the cases where LLaVa predicted that triads were balanced, it generally offered a perplexing or inaccurate reasoning. For example, in several cases LLaVa stated that triads were balanced because "The blue and red lines are of equal length, indicating a balance between the two relationships". One of its most frequent (faulty) justifications for classifying a triad as unbalanced, accurately or inaccurately, was some variation of "the blue and red lines are not parallel, indicating an imbalance in the relationships (Table 2)."

Table 2. Triadic Balance Results: b denotes the number of 'like' (blue) edges. 3b and 0b report accuracy only on triads containing 3 and 0 (+) relations respectively. 1b and 2b contain accuracy metrics on the triads containing 1 and 2 (+) relations respectively. The balanced column contains accuracy calculated on all balanced triads (3b and 1b), unbalanced is calculated on all unbalanced triads (b0 and 2b) and overall is accuracy calculated over all triads

		Accuracy						
		3b	1b	b0	2b	balanced	unbalanced	overall
GPT-4	No Definitions	0.70	**0.50**	0.80	0.37	**0.55**	0.46	**0.51**
	Definitions	**1**	0.20	0.90	0.47	0.40	0.58	0.49
LLaVa	No Definitions	0.10	0.23	0.80	0.73	0.20	0.75	0.48
	Definitions	0	0	**1**	**1**	0	**1**	0.50

5.3 Component Task Results

The top performing model across all evaluation metrics for both the component and isolate counting tasks was GPT-4 with no definitions included in the prompt. However, we note that the inclusion of definitions resulted in a substantial improvement of LLaVa's MSE, which decreased from over 20,000 to below 11. GPT-4 performed relatively well on the isolate counting task, and provided the correct number of isolates 67 of 100 times. We provide results in Table 3.

Table 3. Component Results. We report Accuracy, MAE, and MSE for each model and prompt condition for both the Component counting and Isolate counting tasks

		Components			Isolates		
		Accuracy	MAE	MSE	Accuracy	MAE	MSE
GPT-4	No Definitions	**0.39**	**1.25**	**3.41**	**0.67**	**0.51**	**0.99**
	Definitions	0.37	1.45	4.97	0.64	0.58	1.3
LLaVa	No Definitions	0.03	59.79	23297.95	0.08	32.43	20652.51
	Definitions	0.05	2.81	10.57	0.17	1.27	2.23

6 Discussion and Limitations

Given GPT-4's strong performance on professional exams like the LSAT and MCAT, it is, at least from a human perspective, surprising that it would struggle with something as simple as counting specific elements of graphs. This phenomenon likely relates, at least in part, to how LVMs process images as patches [14]. Additionally, it's unsurprising that GPT-4 would outperform LLaVa given LlaVa almost certainly contains far fewer (3B) parameters than GPT-4[4]. Nonetheless, more research is needed to understand why LVMs struggle on tasks this simple, and future research could explore the performance of LVMs fine-tuned on graph-related tasks. Additionally, the ways that graph visualization parameter selection and prompt engineering impact LVM performance on VNA tasks are clear and important avenues for future research.

7 Conclusion

We propose the task of zero-shot Visual Network Analysis to evaluate the performance of LVMs on graph analytics tasks. We create a benchmark that includes 5 tasks related to 3 core network science concepts: maximum degree, structural balance, and identifying components. We find that across all tasks, LLMs struggled to identify and count the appropriate element of graphs—an essential skill in analyzing network data. We publicly release all generated data and ground-truth labels.

Acknowledgements. The research for this paper was supported in part by the ARMY Scalable Technologies for Social Cybersecurity and the Office of Naval Research, MURI: Persuasion, Identity, & Morality in Social-Cyber Environments under grants W911NF20D0002 and N000142112749. It was also supported by the center for Informed Democracy and Social-Cybersecurity (IDeaS) and the center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. The views and conclusions are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARMY, the ONR, or the US Government.

[4] the exact number of parameters in GPT-4 is not currently known, but it is likely far larger.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Achiam, J., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Brodersen, P.J.N.: Netgraph: publication-quality network visualisations in python. J. Open Source Softw. **8**(87), 5372 (2023). https://doi.org/10.21105/joss.05372
3. Cartwright, D., Harary, F.: Structural balance: a generalization of Heider's theory. Psychol. Rev. **63**(5), 277 (1956)
4. Chen, Z., et al.: Exploring the potential of large language models (llms) in learning on graphs. ACM SIGKDD Explor. Newsl **25**(2), 42–61 (2024)
5. Fatemi, B., Halcrow, J., Perozzi, B.: Talk like a graph: encoding graphs for large language models. arXiv preprint arXiv:2310.04560 (2023)
6. Goh, G., et al.: Multimodal neurons in artificial neural networks. Distill (2021). https://doi.org/10.23915/distill.00030. https://distill.pub/2021/multimodal-neurons
7. Harary, F.: On the notion of balance of a signed graph. Mich. Math. J. **2**(2), 143–146 (1953)
8. Heider, F.: Attitudes and cognitive organization. J. Psychol. **21**(1), 107–112 (1946)
9. Islam, A., Biswas, M.R., Zaghouani, W., Belhaouari, S.B., Shah, Z.: Pushing boundaries: exploring zero shot object classification with large multimodal models. In: 2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 1–5. IEEE (2023)
10. Li, Y., Hu, B., Shi, H., Wang, W., Wang, L., Zhang, M.: Visiongraph: leveraging large multimodal models for graph theory problems in visual context. arXiv preprint arXiv:2405.04950 (2024)
11. Liu, H., et al.: Llava-next: improved reasoning, ocr, and world knowledge (2024). https://llava-vl.github.io/blog/2024-01-30-llava-next/
12. Wasserman, S., Faust, K.: Social network analysis: methods and applications (1994)
13. Wei, Y., Fu, S., Jiang, W., Kwok, J.T., Zhang, Y.: Rendering graphs for graph reasoning in multimodal large language models. arXiv preprint arXiv:2402.02130 (2024)
14. Xu, J., Le, H., Samaras, D.: Zero-shot object counting with language-vision models. arXiv preprint arXiv:2309.13097 (2023)
15. Zeng, J., et al.: Large language models for social networks: applications, challenges, and solutions. arXiv preprint arXiv:2401.02575 (2024)
16. Zhang, C., Wang, S.: Good at captioning, bad at counting: benchmarking gpt-4v on earth observation data. arXiv preprint arXiv:2401.17600 (2024)
17. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. (2024)