GRAPHGPT-O: Synergistic Multimodal Comprehension and Generation on Graphs

Yi Fang*
New York University Shanghai
yf2722@nyu.edu

Bowen Jin*
University of Illinois at Urbana-Champaign
bowenj4@illinois.edu

Jiacheng Shen*
New York University Shanghai
js12556@nyu.edu

Sirui Ding
University of California San Francisco
sirui.ding@ucsf.edu

Qiaoyu Tan
New York University Shanghai
qiaoyu.tan@nyu.edu

Jiawei Han
University of Illinois at Urbana-Champaign
hanj@illinois.edu

Abstract

The rapid development of Multimodal Large Language Models (MLLMs) has enabled the integration of multiple modalities, including texts and images, within the large language model (LLM) framework. However, texts and images are usually interconnected, forming a multimodal attributed graph (MMAG). It is underexplored how MLLMs can incorporate the relational information (i.e., graph structure) and semantic information (i.e., texts and images) on such graphs for multimodal comprehension and generation. In this paper, we propose GRAPHGPT-O*, which supports omni-multimodal understanding and creation on MMAGs. We first comprehensively study linearization variants to transform semantic and structural information as input for MLLMs. Then, we propose a hierarchical aligner that enables deep graph encoding, bridging the gap between MMAGs and MLLMs. Finally, we explore the inference choices, adapting MLLM to interleaved text and image generation in graph scenarios. Extensive experiments on three datasets from different domains demonstrate the effectiveness of our proposed method. Datasets and codes will be publicly available at https://github.com/YiFang99/GraphGPT-O.*

1. Introduction

Multimodal Large Language Models (MLLMs) [7, 21, 26, 35] have made significant progress in recent years, allowing the comprehension and generation of diverse data modalities including text and images. However, in real-world scenarios, there exists a pervasive *graph-structured relationships* between texts and images. Such graph-structured relationship can be described as "*Multimodal Attributed Graphs*" (MMAGs) [16, 31, 44], where nodes are associated with image and text information. For example, the artwork graph [29] is composed of nodes that include images (pictures) and text (titles), with edges representing shared genres and authorship. This structure uniquely represents each artwork in relation to thousands of others within the graph, providing a context that extends beyond simple language descriptions or image references. While MLLMs have demonstrated outstanding comprehension and generation capability for text and image data, it is questionable how they could utilize the structural information on MMAGs.

In this context, we formulate the problem of *multimodal content generation* on MMAGs which tasks MLLMs with producing both a textual description and an accompanying image for a new node based on the graph connectivity and node attributes. This task focuses on generating text-image pairs for a node from MMAGs, reflecting a wide range of practical applications. For example, generating an image and a text for a product node linked to others through co-purchase edges in an e-commerce MMAG is equivalent to recommending [6, 28] potential future products to users. Likewise, creating an image and a title for a virtual artwork node in the art MMAG is comparable to creating virtual artwork [8, 15] that reflects the subtle styles of various artists and genres.

However, directly adopting MLLMs on MMAGs for multimodal content generation presents several challenges:

*Equal contribution.

(1) *Graph Size Explosion*: Although MMAGs provide substantial context for image and text generation, inputting the entire local subgraph structure to a model is impractical due to the exponential increase in size with additional hops, leading to excessively long context sequences. (2) *Non-Euclidean Nature*: Unlike texts or images, which follow linear structures, graphs are inherently non-Euclidean with complex topologies [2], making them challenging to feed into MLLMs. (3) *Hierarchical Modality Dependency*: At the node level, complementary information from associated text and image data enhances the semantic understanding of individual nodes. At the subgraph level, integrated features derived from node text/image semantics and local graph structure enable a more nuanced understanding of the subgraph's context for target node generation. (4) *Inference Dependency*: Due to the intrinsic interdependence between text and image features within a node, as well as the dual objectives of image and text generation, the order of inference across these modalities is critical.

To address these challenges, we introduce GRAPHGPT-O, a multimodal large language model tailored for comprehensive understanding and creation within MMAGs. Our approach features several key contributions: (1) We develop a personalized PageRank-based graph sampling method to extract relevant subgraph information, effectively mitigating the *Graph Size Explosion* issue. (2) We investigate various design approaches for graph linearization, adapting its *Non-Euclidean Nature* to fit a sequential MLLM processing paradigm. (3) We construct a hierarchical graph aligner, incorporating a node-level modality fusion Q-Former and a graph structure Q-Former to capture *Hierarchical Modality Dependency* within MMAGs. (4) We explore different inference strategies, including sequential and parallel generation, to address *Inference Dependency* across modalities in MMAGs. With adaptive graph prompt designs and specialized alignment techniques, GRAPHGPT-O achieves effective comprehension and content generation in MMAGs, overcoming key challenges related to graph topology and multimodal attribute integration.

The primary contributions of this paper are as follows:

- **Problem Formulation and Benchmarking.** We formally define the task of multimodal content generation from MMAGs and introduce three real-world benchmark datasets across domains such as art and e-commerce to support this task.
- **Proposed Methodology.** We introduce GRAPHGPT-O, a multimodal large language model designed to effectively encode graph structures for concurrent image and text generation.
- **Experiments and Evaluation.** We perform comprehensive experiments and evaluations, demonstrating that GRAPHGPT-O achieves significant improvements over baseline models.

2. Problem Formulation

2.1. Multimodal Attributed Graphs

Definition 1 *(Multimodal Attributed Graphs (MMAGs) [16]) A multimodal attributed graph is defined as* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P}, \mathcal{D})$*, where* $\mathcal{V}$*,* $\mathcal{E}$*,* $\mathcal{P}$*, and* $\mathcal{D}$ *denote the sets of nodes, edges, images, and documents, respectively. Each node* $v_i \in \mathcal{V}$ *contains corresponding image information* $p_{v_i} \in \mathcal{P}$ *and textual information* $d_{v_i} \in \mathcal{D}$*.*

Some examples of MMAGs include (1) e-commerce produce graphs ($\mathcal{G}$), where product nodes ($v \in \mathcal{V}$) are associated with product image ($p \in \mathcal{P}$) and title ($d \in \mathcal{D}$); and (2) artwork graphs ($\mathcal{G}$), where artwork nodes ($v \in \mathcal{V}$) contain picture ($p \in \mathcal{P}$) and title ($d \in \mathcal{D}$).

2.2. Problem Definition

Definition 2 *(Node Multimodal Content Generation on MMAGs) In a multimodal attributed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P}, \mathcal{D})$*, given a node* $v_i \in \mathcal{V}$ *within the graph* $\mathcal{G}$*, the goal is to generate* p_{v_i} *and* d_{v_i}*, the corresponding image and text at* v_i*, with a learned model* $(\hat{p}_{v_i}, \hat{d}_{v_i}) = f(v_i, \mathcal{G})$*.*

This problem has numerous real-world applications. In the context of e-commerce, this translates to generating an image (p_{v_i}) and a title (d_{v_i}) for a product (v_i) based on a user's purchase history ($\mathcal{G}$), framing it as a generative recommendation task. In the art domain, this involves generating an image (p_{v_i}) and a title (d_{v_i}) for an artwork (v_i) based on the associated artist's style or genre ($\mathcal{G}$), positioning it as a virtual artwork creation task.

3. Methodology

In this section, we present our GRAPHGPT-O framework, a novel approach for generating image-text pairs on MMAGs using multimodal LLMs (MLLMs). We begin by introducing graph information into MLLMs in Section 3.1. Next, we describe a personalized PageRank-based graph sampling strategy in Section 3.2, addressing the *Graph Size Explosion* challenge. In Section 3.3, we propose graph linearization strategies and develop a hierarchical graph aligner to address the *Non-Euclidean Nature* of graphs and capture *Hierarchical Modality Dependency* in MMAGs. Finally, in Section 3.4, we explore different generation strategies to manage *Inference Dependency* across modalities.

3.1. Multimodal LLM on MMAGs

DreamLLM. The proposed GRAPHGPT-O is built upon DreamLLM [7], an MLLM capable of comprehension and generation on both text and image modalities. To be specific, DreamLLM represents both texts and images as tokens and conducts encoding and generation in an autore-

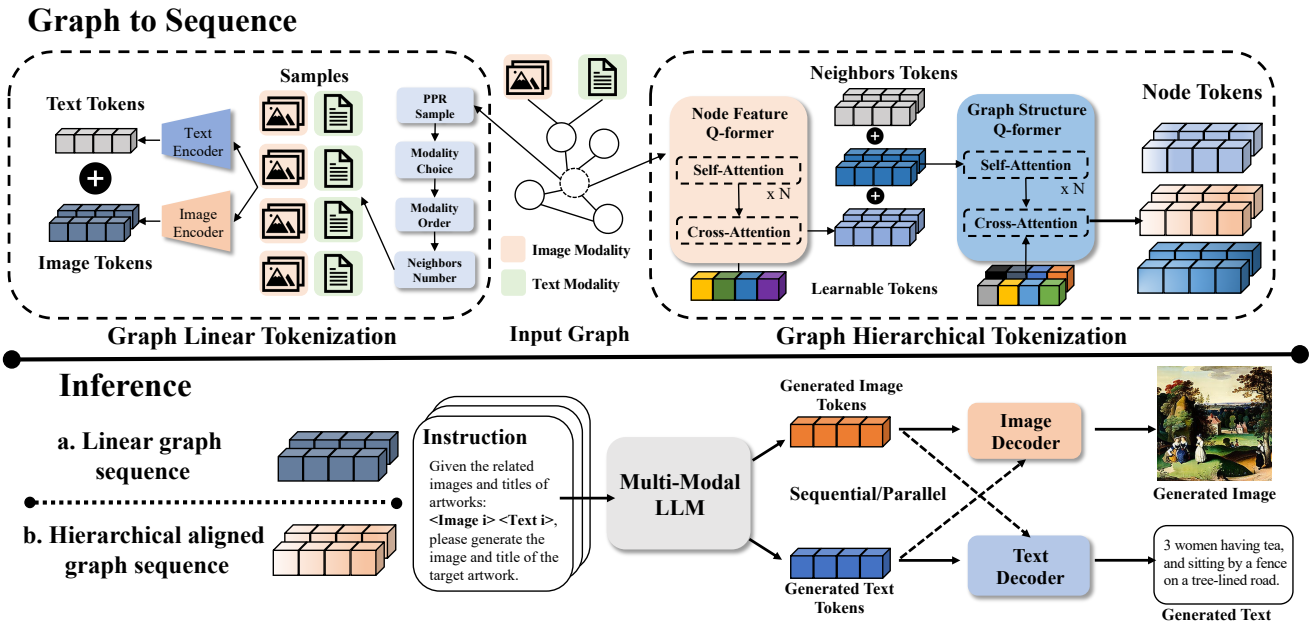

Figure 1. The overall framework of the proposed GRAPHGPT-O is as follows. Given a target node in a multimodal attribute graph (MMAG), we begin by using personalized PageRank for neighbor sampling. These sampled neighboring nodes are then fed into a Hierarchical Multimodal Aligner, which aligns text, image, and graph structure data. Each modality of a node is initially encoded and fused through multiple self-attention and cross-attention layers to produce multimodal node tokens. Subsequently, the tokens are processed by a graph structure Q-former, ultimately serving as inputs to the Multimodal LLM.

gressive fashion:

$$\mathcal{L}_{\text{MLLM}}^{\text{DreamLLM}} = -\mathbb{E}_t \left[\log p \left(x_t^{\text{WI}} \mid \boldsymbol{x}_{<t}^{\text{WI}} \right) \right], \quad (1)$$

where $\boldsymbol{x}^{\text{WI}} = \{x_t^{\text{WI}}\}_{t=1}^{T}$ is an interleaved sequence containing both word tokens $\boldsymbol{w} = \{w_i\}_{i=1}^{N}$ and image tokens $\boldsymbol{I} = \{I_k\}_{k=1}^{K}$. The generated image tokens are decoded into images with a Stable Diffusion [34] trained by the following objective:

$$\mathcal{L}_{\text{SD}}^{\text{DreamLLM}} = \mathbb{E}_{t,\epsilon} \left[\left| \epsilon - \epsilon_\theta \left(z_t; C_{\text{DreamLLM}}^{\text{WI}}, t \right) \right|^2 \right], \quad (2)$$

where $C_{\text{DreamLLM}}^{\text{WI}}$ represents the condition incorporating information from both text and image modalities.

GRAPHGPT-O: introducing graph signals into MLLMs. In the context of MMAGs, generating image and text content for a node v_i requires utilizing semantic information from the surrounding nodes within the *graph* structure. Therefore, we introduce an auxiliary set of graph tokens $\boldsymbol{g}_{v_i} = \{g_j\}_{j=1}^{M}$ as input to the MLLM in addition to the text and image tokens:

$$\mathcal{L}_{\text{MLLM}}^{\text{GRAPHGPT-O}} = -\mathbb{E}_t \left[\log p \left(x_t^{\text{WIG}} \mid \boldsymbol{x}_{<t}^{\text{WIG}} \right) \right], \quad (3)$$

where $\boldsymbol{x}^{\text{WIG}} = \{x_t^{\text{WIG}}\}_{t=1}^{T}$ is an interleaved sequence containing both word tokens $\boldsymbol{w}$, image tokens $\boldsymbol{I}$ and graph tokens $\boldsymbol{g}$. The image decoder Stable Diffusion [34] is then trained by the following objective:

$$\mathcal{L}_{\text{SD}}^{\text{GRAPHGPT-O}} = \mathbb{E}_{t,\epsilon} \left[\left| \epsilon - \epsilon_\theta \left(z_t; C_{\text{GRAPHGPT-O}}^{\text{WIG}}, t \right) \right|^2 \right], \quad (4)$$

where $C_{\text{GRAPHGPT-O}}^{\text{WIG}}$ represents the condition incorporating information from text, image, and graph modalities.

3.2. Personalized PageRank Neighbor Sampling.

A simple approach to obtain $\boldsymbol{g}_{v_i}$ is to encode the entire local subgraph of v_i within $\mathcal{G}$. However, this becomes impractical as the subgraph size grows exponentially with each additional hop, resulting in excessively long context sequences. Additionally, irrelevant or extraneous information in the local subgraph could misguide the model. To overcome this, inspired by [11], we utilize personalized PageRank (PPR) to selectively gather information for constructing $\boldsymbol{g}_{v_i}$ from a graph structure perspective.

To be specific, PPR [13] utilizes the graph structure to produce a ranking score, $P_{i,j}$, for each node v_j relative to a target node v_i. A higher score $P_{i,j}$ indicates a stronger "similarity" or relevance between nodes v_i and v_j. We represent the PPR scores across all nodes with the PPR matrix $\boldsymbol{P} \in \mathbf{R}^{n \times n}$, where each row $P_{i,:}$ corresponds to the PPR vector for node v_i. The PPR matrix $\boldsymbol{P}$ is computed by solving the following equation:

$$\boldsymbol{P} = \beta \hat{\boldsymbol{A}} \boldsymbol{P} + (1 - \beta) \boldsymbol{I}. \quad (5)$$

where β is the reset probability governing the random walk in PPR and $\hat{\boldsymbol{A}}$ is the normalized adjacency matrix. Using the PPR matrix, we define the PPR-based graph neighborhood nodes $N(v_i)$ to calculate $\boldsymbol{g}_{v_i}$ as the top K most relevant neighbors, obtained by maximizing the sum of PPR scores for the selected neighbors:

$$N(v_i) = \underset{N(v_i)\subset\mathcal{V},|N(v_i)|=K}{\operatorname{argmax}} \sum_{v_j\in N(v_i)} P_{i,j}. \tag{6}$$

This selection ensures that $N(v_i)$ captures the nodes most similar to v_i, based on their PPR scores.

3.3. Multimodal Graph as Sequence

After obtaining $N(v_i)$, the problem is how to extract meaningful graph representations from it. Given that GRAPHGPT-O takes sequential data as input, this involves tokenizing $N(v_i)$ into sequence $\boldsymbol{g}_{v_i}$. Previous studies [27, 40, 45] have explored methods for inputting text-attributed graphs as sequences into LLMs, but handling multimodal attributed graphs presents greater complexity. In this section, we explore two ways to achieve this including (1) Linearization: simply linearizing the textual and image features in $N(v_i)$ into a sequence, and (2) Hierarchical Aligner: a hierarchical graph encoder to obtain deep representations as tokens for $N(v_i)$.

3.3.1 Graph Linear Tokenization

We first discuss tokenizing $N(v_i)$ with simple sequence linearization. This involves designing rules Linearize$(\cdot)$ to transform textual and image features in $N(v_i)$ into $\boldsymbol{g}_{v_i}$:

$$\boldsymbol{g}_{v_i} = \text{Linearize}(N(v_i)) \tag{7}$$

Given that $N(v_i)$ is a set of nodes and each $v_j \in N(v_i)$ is associated with both text information d_{v_j} and image information p_{v_j}, the design of the linearization rule should consider three factors: (1) modality choice; (2) modality order and (3) number of neighbors, which are discussed as follows:

Modality choice. Depending on the graph, it is possible that presenting only texts $\{d_{v_j}|v_j \in N(v_i)\}$ or only images $\{d_{v_j}|v_j \in N(v_i)\}$ or both of them could benefit the multimodal content generation on MMAGs.

Modality order. Given that we have both text and image modality, it is flexible to adjust the order of different information, including (1) all images first, followed by texts, (2) all texts first, followed by images, and (3) interleaving image and text for each node $v_j \in N(v_i)$.

Number of neighbors. $N(v_i)$ is a list of nodes ranked by PPR score. Including more neighbors $v_j \in N(v_i)$ into $\boldsymbol{g}_{v_i}$ could potentially add more information but at the same time increase noise.

In Section 4.2, we conduct systematic experiments on how different design choices affect the model performance. After the design choice is given, $\{d_{v_j}|v_j \in N(v_i)\}$ are tokenized with text tokenizer and $\{d_{v_j}|v_j \in N(v_i)\}$ are tokenized with the pretrained CLIP encoder [32] similar to [7].

3.3.2 Graph Hierarchical Tokenization

Although linearization offers a solution for graph tokenization, it fails to capture hierarchical modality dependencies in MMAGs. To be specific, at the node level, the combined information from associated text and image data contributes to a richer semantic representation of individual nodes. At the subgraph level, features synthesized from node-level semantics, alongside the local graph structure, enable a more comprehensive contextual understanding, thereby enhancing the generation of target nodes. To this end, we design a hierarchical aligner $\mathcal{F}(\cdot)$ with a node feature Q-Former $\phi(\cdot)$ and a graph structure Q-Former $\psi(\cdot)$ to capture the node-level and subgraph-level modality dependency respectively:

$$\boldsymbol{g}_{v_i} = \mathcal{F}(N(v_i)) = \psi(\{\phi(v_j)|v_j \in N(v_i)\}) \tag{8}$$

Node Feature Q-Former. It is proposed to learn node representations for $v_j \in N(v_i)$ considering the node-level modality dependency. As shown in Figure 1, the Q-Former comprises two core Transformer [38] modules motivated by [21]: (1) a self-attention module that facilitates deep information exchange between node text features and image features; (2) a cross-attention module that compresses node feature into a fixed number of representations.

The associated text d_{v_j} and image p_{v_j} of a node $v_j \in N(v_i)$ are first transformed into token representations $\boldsymbol{w}_{v_j}$ and $\boldsymbol{I}_{v_j}$ with text tokenizer and pretrained CLIP encoder respectively, which are then concatenated to form the initial input embedding:

$$\mathbf{H}_{v_j}^{(0)} = \left[\boldsymbol{w}_{v_j}; \boldsymbol{I}_{v_j}\right] \in \mathbb{R}^{d\times(|d_{v_j}|+|p_{v_j}|)} \tag{9}$$

The self-attention Transformer layers are designed to perform text and image modality information exchange calculated by:

$$\mathbf{H}_{v_j}^{(t)} = \text{Trans}_{\text{SAT}}\left(q, k, v = \mathbf{H}_{v_j}^{(t-1)}\right) \tag{10}$$

Following L_1 self-attention Transformer layers, a cross-attention Transformer layer is applied, extracting the core feature into a fixed number of representations:

$$\mathbf{H}_{v_j} = \text{Trans}_{\text{CAT}}\left(q = \boldsymbol{Q}_V; k, v = \mathbf{H}_{v_j}^{(L_1)}\right) \tag{11}$$

where $\boldsymbol{Q}_V \in \mathcal{R}^x$ is a node-level information aggregation soft prompt. The final representation $\mathbf{H}_{v_j}$ is leveraged as modality fused node feature representation.

Graph Structure Q-Former. It is designed to aggregate the local context semantics inside $N(v_i)$, capturing the subgraph level modality dependency. Similar to node feature Q-Former, graph structure Q-Former also contains two core Transformer modules: (1) a self-attention module that enables deep information integration inside the local subgraph; (2) a cross-attention module that aggregates the local semantics into a fixed number of representations.

The node representations $\mathbf{H}_{v_j}$ for $v_j \in N(v_i)$ obtained from the node feature Q-Former are concatenated and serve as the initial inputs to the graph structure Q-Former:

$$\mathbf{G}^{(0)}_{N(v_i)} = \left[\mathbf{H}_{v_j} \mid v_j \in N(v_i)\right] \tag{12}$$

The self-attention Transformer layers are then applied to conduct deep information fusion between nodes inside the local subgraph:

$$\mathbf{G}^{(t)}_{N(v_i)} = \text{Trans}_{\text{SAT}}\left(q, k, v = \mathbf{G}^{(t-1)}_{N(v_i)}\right) \tag{13}$$

After the L_2 self-attention Transformer layers, a cross-attention Transformer layer is designed to compress essential local graph features into a fixed set of representations:

$$\mathbf{G}_{N(v_i)} = \text{Trans}_{\text{CAT}}\left(q = \boldsymbol{Q}_G; k, v = \mathbf{G}^{(L_2)}_{N(v_i)}\right) \tag{14}$$

where $\boldsymbol{Q}_G \in \mathcal{R}^x$ is a subgraph-level information aggregation soft prompt. The final representation is leveraged as graph token representations which are inputted into the MLLM: $\boldsymbol{g}_{v_i} = \mathbf{G}_{N(v_i)}$.

3.4. Inference Strategy

Given the inherent interdependence between textual and visual features (d_{v_i} and p_{v_i}) within a node v_i and the joint objectives of generating both image and text, the order of inference across these modalities plays a crucial role. To this end, we propose two strategies: (1) sequential inference and (2) parallel inference.

Sequential Inference. The proposed framework employs a sequential dual-generation process, in which one modality is generated first and subsequently serves as a conditioning factor for the generation of the other modality. Specifically, this approach enables us to generate text d_{v_i} by optimizing $p(d_{v_i}|g_{v_i})$ and then generate the corresponding image p_{v_i} by maximizing $p(p_{v_i}|g_{v_i}, d_{v_i})$. Alternatively, we can initiate generation with the image p_{v_i} by maximizing $p(p_{v_i}|g_{v_i})$ and then produce the text d_{v_i} by optimizing $p(d_{v_i}|g_{v_i}, p_{v_i})$. This sequential conditioning strategy ensures that the second generation step is contextually anchored in the outcome of the first, potentially enhancing coherence and consistency across modalities.

Parallel Inference. The framework is designed to enable simultaneous dual generation of text and image by jointly optimizing $p(d_{v_i}|g_{v_i})$ and $p(p_{v_i}|g_{v_i})$. This concurrent generation approach allows the production of d_{v_i} and p_{v_i} to proceed independently, mitigating the risk of error propagation from one modality serving as a conditional input for the other. Consequently, this parallel optimization strategy can reduce dependency on sequential conditioning, enhancing robustness in the generation process.

4. Experiment

4.1. Experimental Setups

We conduct experiments on three multimodal attributed graphs from e-commerce domain and art domain. Our quntitive evaluation utilized CLIP scores, text perplexity, as well as KL divergence. Detailed information about datasets and metrics are provided in the appendix.

4.2. Graph Linear Tokenization

In this section, we study the quantitative results with graph linear tokenization, which are presented in Table 1, from which we observe the following:

(1) Node Modality Integration. Utilizing both modalities together generally improves model performance, indicating that integrating multiple information sources leads to a more comprehensive understanding of the data.

(2) Node Modality Order. The order in which the modalities are processed does not consistently or significantly affect model performance.

(3) Inference Strategy. Generating the image first typically enhances the quality of the synthesized image but may reduce text quality, whereas starting with text generation results in the lowest KL-DV score.

4.3. Graph Hierarchical Tokenization

4.3.1 Quantitative Evaluation.

In this section, we compare the results of the original DreamLLM [7] and Chameleon [37], and DreamLLM fine-tuned with graph linear tokenization prompts named GRAPHGPT-O (Hard), and trained with an additional hierarchical aligner module named GRAPHGPT-O (soft). The default prompt setting for training and inference utilizes both modalities, with text-first in the instruction and generating text-first during inference. The results are shown in Table 2, from which we can observe that GRAPHGPT-O (soft) outperforms baselines in most cases, especially aligns better with the golden sub-graph.

4.3.2 Qualitative Evaluation.

We performed a qualitative evaluation by randomly selecting several generated cases, and comparing them with ground-truth, DreamLLM, and ChatGPT-4o. The results are presented in Figure 4, which includes sampled neighbor

Modality	Order	Inference	ART500K				Beauty				Baby			
			CLIP-I2	Perplexity	CLIP-IT	KL-DV	CLIP-I2	Perplexity	CLIP-IT	KL-DV	CLIP-I2	Perplexity	CLIP-IT	KL-DV
Text-only	Text-first	Text-first	65.83	163.3	22.66	4.65	55.49	193.8	24.6	10.72	**78.89**	328.3	17.88	2.51
		Image-first	65.31	619.9	24.54	5.04	65.56	668.5	**25.64**	14.72	75.36	819.8	23.84	1.32
		Parallel	65.31	158.5	16.37	9.96	65.56	206.9	19.99	22.31	75.36	253.5	18.74	6.99
Image-only	Image-first	Text-first	73.08	130.4	19.53	0.33	62.25	124.5	9.49	18.55	72.40	155.5	27.8	**0.73**
		Image-first	75.55	177.7	12.18	9.85	**67.61**	266.8	20.85	14.83	75.22	**130.1**	10.62	2.54
		Parallel	75.55	460.8	22.66	5.76	**67.61**	**108.8**	25.48	10.82	75.22	178.8	21.76	1.14
Text-Image	Text-first	Text-first	71.15	555.7	18.86	0.49	60.5	514.7	20.84	**9.85**	67.98	402.3	27.59	1.93
		Image-first	**79.26**	**117.7**	20.15	2.78	66.89	379.7	6.78	10.64	59.27	153.3	8.78	16.07
		Parallel	79.26	737.7	22.56	3.82	66.89	407.3	19.50	10.65	59.27	839.2	14.62	7.43
	Image-first	Text-first	74.14	217.3	23.81	**0.19**	62.19	259.3	22.83	10.27	71.38	325.4	**33.31**	5.86
		Image-first	77.81	437.8	19.32	3.19	60.55	353.7	14.32	24.77	65.48	242.2	9.62	1.15
		Parallel	77.81	219.8	22.18	2.97	60.55	207.1	22.51	22.21	65.48	169.9	23.42	0.79
	Interleaved	Text-first	68.40	315.7	18.57	0.70	64.70	310.8	25.5	10.39	71.71	522.5	31.86	0.79
		Image-first	77.71	117.9	20.84	2.66	64.64	346.5	6.79	10.63	62.62	572.5	21.98	0.88
		Parallel	77.71	402.8	**28.41**	2.68	64.64	354.8	24.69	18.38	62.62	572.5	13.95	7.72

Table 1. Evaluation Results for Different Modalities and Orders on ART500K, Amazon-Beauty, and Amazon-Baby Datasets

Model	ART500K				Beauty				Baby			
	CLIP-I2	Perplexity	CLIP-IT	KL-DV	CLIP-I2	Perplexity	CLIP-IT	KL-DV	CLIP-I2	Perplexity	CLIP-IT	KL-DV
Janus	59.32	351.2	21.43	7.59	42.52	415.6	17.8	1.45	52.81	324.5	25.6	2.43
Emu3	62.11	257.5	20.07	9.83	45.82	398.3	24.2	5.96	58.8	374.5	23.7	3.55
Chameleon	61.19	228.3	23.87	4.18	43.73	180.9	16.6	22.80	45.20	144.7	0.56	1.87
DreamLLM	71.15	555.7	18.86	0.4882	60.5	514.7	20.84	9.85	67.98	402.3	27.59	1.92
GRAPHGPT-O (Hard)	**77.62**	347.4	18.58	0.9377	57.99	**107.9**	24.66	12.75	68.23	124.8	20.24	1.39
GRAPHGPT-O (Soft)	72.64	**59.8**	**25.63**	**0.4327**	**63.46**	285.0	**27.38**	**5.82**	**74.77**	**103.2**	**31.14**	**0.23**

Table 2. Results for different backbones on ART500K, Amazon-Beauty, and Amazon-Baby Datasets

	CLIP-I2	Perplexity	CLIP-IT	KL-DV
GRAPHGPT-O w/o GSQ	61.44	**59.67**	**26.96**	9.1406
GRAPHGPT-O w/o NFQ	72.60	71.96	26.74	2.5050
GRAPHGPT-O	**72.64**	59.80	25.63	**0.4327**

Table 3. The impact of different modules in GRAPHGPT-O. w/o GSQ means without graph structure Q-Former and w/o NFQ means without node feature Q-Former.

	CLIP-I2	Perplexity	CLIP-IT	KL-DV
GRAPHGPT-O with GNN	65.53	599.1	22.5	4.83
GRAPHGPT-O with GSQ	**71.15**	**555.7**	**18.86**	**0.49**

Table 4. The impact of using GNN or graph structure Q-former (GSQ) for structure information learning in GRAPHGPT-O on ART500K dataset.

images and text from the graph alongside the ground truth images and text. These findings show that GRAPHGPT-O generates images that align closer with the contextual information derived from the golden sub-graph, while DreamLLM and ChatGPT- 4o stick to one style and fail to adapt based on the input.

4.4. Ablation Study

First, we evaluate the effectiveness of our **hierarchical aligner module** by individually removing the node feature Q-former and the graph structure Q-former. The results, presented in Table 3, demonstrate that both modules contribute significantly to overall performance. Removing the graph structure causes a substantial increase in the KL-DV score while excluding the node features results in a higher perplexity for text.

To further evaluate the effect of our hierarchical aligner module, a **GNN module** is used to replace it and the results are shown in Table 4, which shows that graph structure q-former is much better.

We then assess the impact of our **Personalized PageRank sampling** method. From Figure 3, it can be observed that our proposed Personalized PageRank sampling strategy effectively captures neighbors that contribute most to the ground truth in terms of texture, artistic style, and visual consistency. This results in a generated image that more closely resembles the ground-truth image's detailed patterns and overall aesthetic.

4.5. Other Studys

Study of generation with partial node feature guidance. We further conduct a study on the performance of GRAPHGPT-O with additional node text guidance or node image guidance. From Figure 4 we can see that the style and the character information is well-captured.

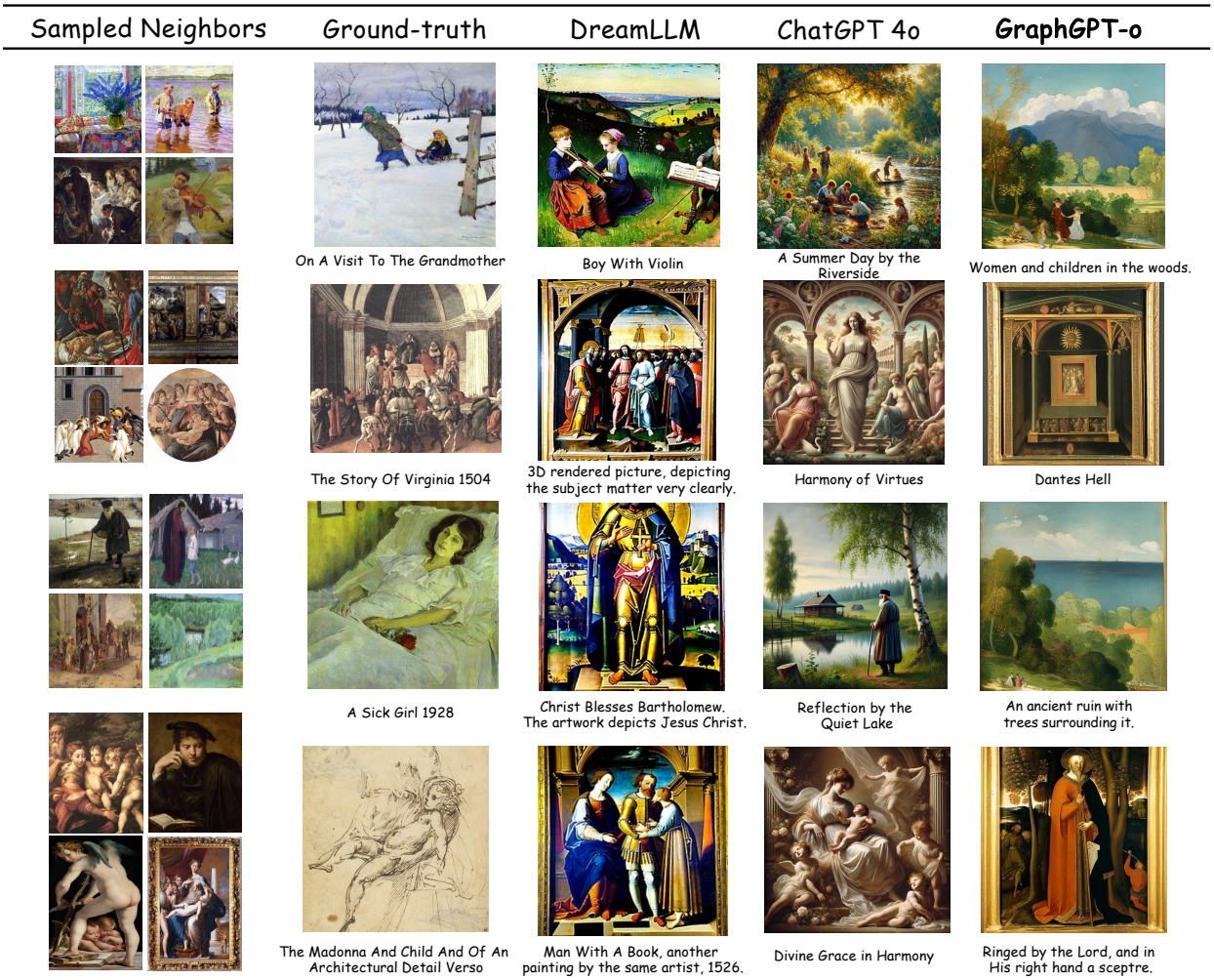

Figure 2. Qualitative evaluation. Our method exhibits better consistency with the ground truth by better utilizing the graph information from neighboring nodes.

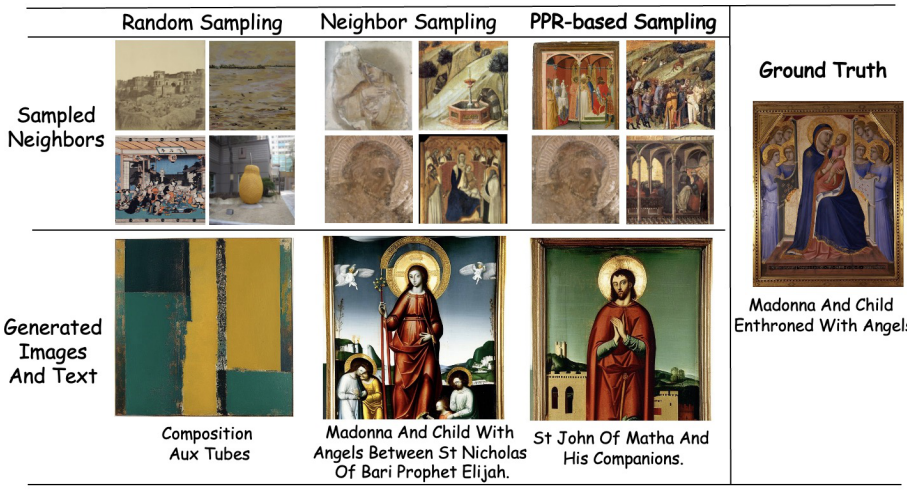

Figure 3. The impact of sampling strategies. Our proposed personalized PageRank sampling strategy leads to better image-text pair.

Figure 4. Study of GRAPHGPT-O generation with auxiliary node feature guidance: either image or text.

5. Related Work

5.1. Large Language Models on Graphs

Large Language Models (LLMs) have driven substantial progress in graph learning applications [18, 33]. Graph data, on the one hand, can be utilized directly to train LLMs [3, 43]. For instance, models like Heterformer [17] and Edgeformer [12] introduce graph-enhanced Transformer architectures, positioning them as foundational models for graph-based LLMs. GraphGPT [36] leverages graph structural data via graph instruction tuning, facilitating robust generalization across supervised and zero-shot graph learning tasks. LLaGA [5] employs a parameter-free GNN and incorporates the graph structure based on the order of node tokens. Similarly, InstructGraph [39] employs a structured format verbalizer to encode graph data, enhancing LLMs in tasks requiring graph reasoning and generation. GraphAdapter [14] incorporates GNNs as efficient adapters for LLMs, while GAugLLM [10] advances self-supervised learning through augmented node features generated by an MoE module, effectively bridging textual and graph structures. UniGLM [9] uses structure information to build positive sample pairs in contrastive learning framework to train a unified text encoder. ZeroG [22] also utilized LLM for graph zero-shot transfer learning. GraphWiz [4] equip LLMs to tackle a broad spectrum of graph problems using explicit reasoning paths. GLBench [23] provides a fair and thorough comparison of different LLM for graphs methods. On the other hand, graph data can be utilized as external knowledge in a plug-and-play manner with LLMs [19, 30]. For example, Graph Chain-of-Thought [19] proposes an iterative framework that enables LLMs to reason, interact, and operate effectively on graphs. GNN-RAG [30] introduces a retrieval-augmented generation framework [20], using a GNN retriever to extract knowledge from graph data. Despite these advances, existing research has primarily focused on graphs with textual attributes, leaving multimodal attributed graphs underexplored. G-Refer [24] proposed a hybrid graph retrieval mechanism to retrieve explicit CF signals from both structural and semantic perspectives

5.2. Multimodal Large Language Models

Multimodal Large Language Models (MLLMs) have advanced the field by enabling unified multimodal understanding and generation within a single autoregressive framework [41, 42]. In terms of multimodal comprehension, models like Flamingo [1] process visual data interleaved with text, utilizing a gated cross-attention layer to encode inputs and produce free-form textual output. BLIP-2 [21] introduces the Q-Former architecture, which maps images into a hidden space aligned with text tokens in LLMs, while LLaVA [25] simplifies this framework further with a projector and explores instruction tuning within the multimodal domain. Despite these advancements, current MLLMs primarily emphasize text generation and lack the capability to synthesize multimodal outputs (*e.g.*, images). To address this, DreamLLM [7] integrates an LLM backbone with a diffusion model to enable image generation as a multimodal output. Emu2 [35] scales this architecture to 37B parameters, demonstrating strong multimodal in-context learning and the ability to handle complex tasks requiring real-time reasoning, such as visual prompting and object-grounded generation. Chameleon [37] proposes a stable training strategy from the ground up, featuring an alignment process and architectural parameterization tailored to early-fusion, token-based, mixed-modal settings. Nevertheless, most existing approaches overlook the relational dynamics between text and images, limiting their applicability to multimodal content generation tasks on multimodal attributed graphs (MMAGs).

6. Conclusions

In this paper, we address the challenge of multimodal content generation on multimodal attributed graphs (MMAGs). To this end, we propose a graph-enhanced multimodal large language model, GRAPHGPT-O, designed with the following components: (1) A personalized PageRank-based sampling strategy to extract informative neighbors from the graph, effectively mitigating the challenge of graph size explosion; (2) A transformation mechanism that encodes graph information as sequences, employing either linearization or deep graph encoding with a hierarchical aligner, thereby addressing the non-Euclidean nature of graphs and hierarchical modality dependencies; (3) Dual inference modes supporting both sequential and parallel inference to alleviate inference dependency issues. We conduct comprehensive experiments on MMAGs within art and e-commerce domains, demonstrating the effectiveness of our approach against strong baseline methods. Future work includes extending MLLMs for discriminative tasks on MMAGs and capturing the complex heterogeneous relations between texts and images within these graphs.

Acknowledgment

This research was supported by Apple PhD Fellowship. The research was supported in part by US DARPA INCAS Program No. HR0011-21-C0165 and BRIES Program No. HR0011-24-3-0325, National Science Foundation IIS-19-56151, the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 8

[2] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 2

[3] Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. Graphwiz: An instruction-following language model for graph problems. *arXiv preprint arXiv:2402.16029*, 2024. 8

[4] Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. Graphwiz: An instruction-following language model for graph problems, 2024. 8

[5] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant, 2024. 8

[6] Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. A review of modern recommender systems using generative models (gen-recsys), 2024. 1

[7] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Dreamllm: Synergistic multimodal comprehension and creation, 2024. 1, 2, 4, 5, 8

[8] Ziv Epstein, John Kowalski, Laura Thomas, and Steve Zhang. Art and the science of generative ai: A deeper dive. *arXiv preprint arXiv:2306.04141*, 2023. 1

[9] Yi Fang, Dongzhe Fan, Sirui Ding, Ninghao Liu, and Qiaoyu Tan. Uniglm: Training one unified language model for text-attributed graph embedding, 2024. 8

[10] Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 747–758, 2024. 8

[11] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank, 2022. 3

[12] Tao Ge, Si-Qing Chen, and Furu Wei. Edgeformer: A parameter-efficient transformer for on-device seq2seq generation. *arXiv preprint arXiv:2202.07959*, 2022. 8

[13] Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*. 3

[14] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can llms effectively leverage graph structural information: when and why. *arXiv preprint arXiv:2309.16595*, 2023. 8

[15] Nisha Huang, Fan Tang, Weiming Dong, and Changsheng Xu. Draw your art dream: Diverse digital art synthesis with multimodal guided diffusion, 2022. 1

[16] Bowen Jin, Ziqi Pang, Bingjun Guo, Yu-Xiong Wang, Jiaxuan You, and Jiawei Han. Instructg2i: Synthesizing images from multimodal attributed graphs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 1, 2

[17] Bowen Jin, Yu Zhang, Qi Zhu, and Jiawei Han. Heterformer: Transformer-based deep node representation learning on heterogeneous text-rich networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1020–1031, 2023. 8

[18] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 8

[19] Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, et al. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*, 2024. 8

[20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. 8

[21] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1, 4, 8

[22] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs, 2024. 8

[23] Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor Wai Kin Chan, and Jia Li. Glbench: A comprehensive benchmark for graph with large language models, 2024. 8

[24] Yuhan Li, Xinni Zhang, Linhao Luo, Heng Chang, Yuxiang Ren, Irwin King, and Jia Li. G-refer: Graph retrieval-augmented large language model for explainable recommendation, 2025. 8

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 8

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1

[27] Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. Can we soft prompt llms for graph learning tasks? 4

[28] Zihan Liu, Yupeng Hou, and Julian McAuley. Multi-behavior generative recommendation, 2024. 1

[29] Hui Mao, James She, and Ming Cheung. Visual arts search on mobile devices. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(2s): 60, 2019. 1

[30] Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*, 2024. 8

[31] Ciyuan Peng, Jiayuan He, and Feng Xia. Learning on multimodal graphs: A survey, 2024. 1

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4

[33] Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6616–6626, 2024. 8

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[35] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024. 1, 8

[36] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024. 8

[37] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 5, 8

[38] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 4

[39] Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. Instructgraph: Boosting large language models via graph-centric instruction tuning and preference alignment. *arXiv preprint arXiv:2402.08785*, 2024. 8

[40] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs, 2024. 4

[41] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 8

[42] Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024. 8

[43] Jiawei Zhang. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*, 2023. 8

[44] Jing Zhu, Yuhang Zhou, Shengyi Qian, Zhongmou He, Tong Zhao, Neil Shah, and Danai Koutra. Multimodal graph benchmark, 2024. 1

[45] Kerui Zhu, Bo-Wei Huang, Bowen Jin, Yizhu Jiao, Ming Zhong, Kevin Chang, Shou-De Lin, and Jiawei Han. Investigating instruction tuning large language models on graphs. *arXiv preprint arXiv:2408.05457*, 2024. 4