

LARGE LANGUAGE MODELS ENHANCE GRAPH LEARNING WITHOUT GRAPH SERIALIZATION

Anonymous authors

Paper under double-blind review

ABSTRACT

We introduce GINAT, a framework that augments graph neural networks (GNNs) with global graph descriptions encoded by large language models (LLMs). While prior work on text-attributed graphs (TAGs) integrates text features tied to nodes or edges, our approach leverages external prompt datasets that describe graph semantics independently of the downstream task. These prompts are embedded by a frozen LLM and injected into the message-passing process via cross-attention, allowing GNNs to incorporate global contextual information during local computations. We provide a theoretical analysis showing that GINAT provably increases the expressive capacity of GNNs while preserving permutation equivariance.

Extensive experiments across diverse graph benchmarks demonstrate consistent improvements over strong GNN baselines, with performance gains of up to 10%. Ablation studies further confirm that these improvements stem from the semantic content of the prompts rather than from random noise or model size. Together, our findings highlight LLM-embedded prompts as a principled and effective new modality for enhancing graph learning.

Our code is available at <https://anonymous.4open.science/r/11m4gfm-3FBE/>

1 INTRODUCTION

What do graphs and natural language have in common? At first glance, they appear to be fundamentally different. Graphs offer a formal framework for decomposing objects, concepts, and complex systems into smaller components and the relationships among them. They enable precise, unambiguous representations that support mathematical analysis and computational processing. Natural language, in contrast, provides a flexible medium for high-level discourse, abstracting away fine details while enriching communication with context, nuance, and cross-domain associations (Gentner & Asmuth, 2019; Williams et al., 2017; Korta & Perry, 2024). In this work, we regard graphs and natural language as *complementary* representational paradigms, linked by a fundamental commonality: both constitute human-engineered abstractions devised to model, organize, and interpret the complexity of the world.

The past decade has seen the rise of deep learning models that broke ground in natural language processing (NLP) and graph processing. Graph neural networks (GNNs) have made tremendous contributions in drug design and related biochemical tasks (Jumper et al., 2021; Vignac et al., 2023), recommendation systems (Wang et al., 2019), among other fields. LLMs de-facto solved most of the NLP tasks, and are widely believed to be the bringers of artificial intelligence and artificial generative intelligence (Qin et al., 2024; Mumuni & Mumuni, 2025).

Encouraged by the success of foundation models in other modalities such as language (Mumuni & Mumuni, 2025) and vision (Awais et al., 2025), the graph learning community has made efforts towards equivalent models for the graph modality. Towards this goal, multiple approaches were proposed; examples include pretraining for GNNs (Hou et al., 2022), mixture-of-experts (Zhao et al., 2024), and LLM-based inference (Liu et al., 2024).

Motivated by these developments, we adopt a novel perspective on the relationship between graphs and natural language. Our approach stems from the following fundamental hypothesis: *Graph inference would benefit from global information about the input graph.* As stated earlier, graph

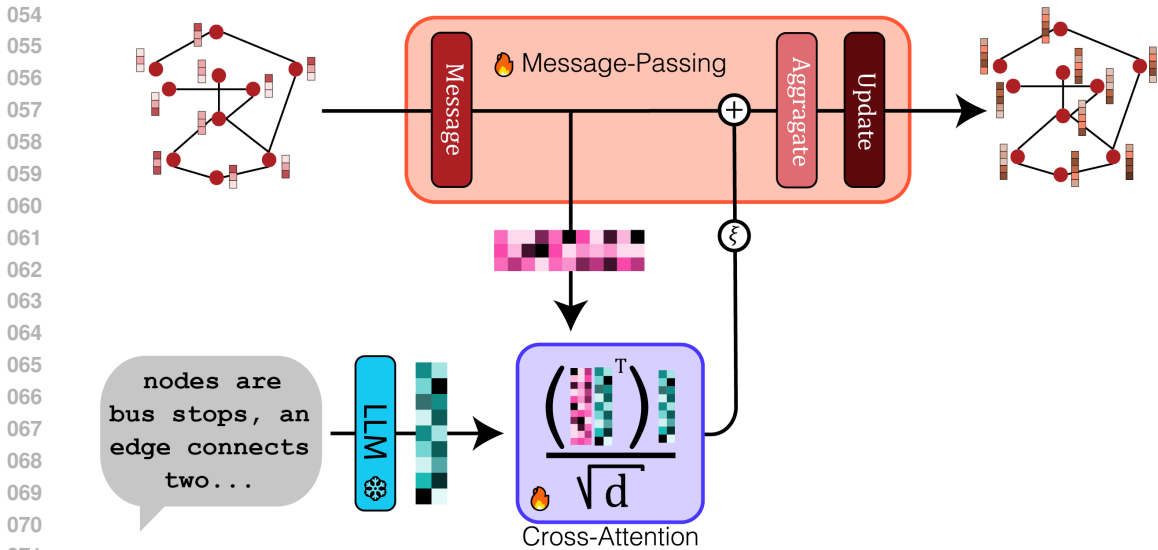


Figure 1: Illustration of GINAT. A graph with initial features is associated with a prompt that describes it. We enrich traditional message passing by cross-attention between graph messages with a representation of its natural language description, generated by a frozen, pretrained LLM. The output of the cross-attention operation is then scaled by a learned parameter ξ and summed with the original messages.

representation of data and textual description complement each other. For example, a cycle in a molecular graph has a different meaning than a cycle in a transportation graph. When humans reason about a graph, they have *prior knowledge* about what it represents, and how to give meaning to substructures. We argue that providing graph models with such prior knowledge is essential towards graph foundation models.

Our Approach. In this work, we augment graphs with textual descriptions. In particular, our approach consists of two parts: data and technique. Given a graph of a certain nature (for example: molecular data, citation networks, social networks and others), we associate the graph with a prompt that describes it. Such a description can be about the graph as a whole (e.g., this graph represents a molecule) or as a sum of its components (e.g., nodes are papers and edges indicate citation). In Figure 3 and Appendix H, we provide concrete examples of such prompts. We note that the prompts we use do not contain information on any downstream task.

Therefore, we call our approach GINAT: **Graph Inference by Natural-language Augmentation of Topology**. Our framework augments message-passing layers with prompt embeddings encoded by a pre-trained LLM, followed by a cross-attention processing step with messages passed on the graph’s edges by said layers. Figure 1 illustrates the approach, and Section 3 provides implementation details. The benefits are twofold: the prompts are enriched by the prior knowledge learned by the LLM, and the GNN receives explicit graph information.

Our contributions are summarized as follows:

- In Section 3, we propose GINAT for graph inference using natural-language data external to the dataset. We also provide the datasets of graph-descriptive prompts.
- In Section 4, we provide extensive theoretical analysis of GINAT, and prove that it improves the capacity of the GNN while maintaining permutation equivariance.
- In Section 5, we establish the potential of the approach. We show improved results on a variety of datasets and using different backbone GNNs.

2 RELATED WORK

In this section, we present ongoing lines of research related to our work.

2.1 TEXT-ATTRIBUTED GRAPHS

Text-attributed graphs (TAGs) combine graph structure with textual node features and appear in many domains, from citation networks to e-commerce. Several recent works advance TAG research. Yan et al. (2023a) introduces a comprehensive benchmark with new datasets, tasks, and systematic evaluation. Yan et al. (2023b) propose GLEM, a scalable framework for large TAGs using variational inference and message passing. Zhang et al. (2024) present the first benchmark for DyTAGs (dynamic TAGs), curating large-scale time-evolving graphs and defining four tasks. Yan et al. (2024) develop an LLM-to-LM Interpreter that distills explanations from LLMs to enrich TAG representations.

While this progress inspired our work, our approach differs fundamentally. TAGs integrate text as node features, whereas we leverage external text that is independent of the target dataset and agnostic to the downstream task, describing instead the semantics of the graph and its sub-entities.

2.2 LARGE LANGUAGE MODELS AS ENHANCERS FOR GRAPH NEURAL NETWORKS

LLMs offer rich semantic priors that can complement the structural focus of GNNs, making them especially relevant for foundation-style graph models. Prior work highlights several roles for LLMs: as *knowledge injectors*, filling gaps in incomplete graphs and improving link prediction through alignment with textual embeddings Li et al. (2023b); Chen et al. (2024a); Tian et al. (2024); as sources of *self-supervised signals*, generating semantic objectives that enrich pretraining beyond structural tasks You et al. (2024); Li et al. (2023b); and as *interpretable enhancers*, producing natural-language rationales that improve transparency and support human-in-the-loop reasoning Gu et al. (2024).

2.3 GRAPH NEURAL NETWORKS AS ENHANCERS FOR LARGE LANGUAGE MODELS

Recent work explores how GNNs enrich LLMs with structured relational reasoning. While LLMs excel at contextual language understanding, they remain limited in explicitly modeling entities and relations. GNNs, with their inductive bias for graph data, have been integrated into LLM-based systems in several ways. One line of work uses knowledge graph augmentation, where GNNs encode entities and relations into embeddings fused with LLM representations. Examples include LinkBERT, which leverages inter-document links during pretraining Yasunaga et al. (2022), and methods that inject graph-based entity embeddings to improve factual QA and reasoning Zhang et al. (2023).

A second line investigates hybrid GNN-LLM architectures. Molecular graph encoders combined with LLMs enhance drug discovery tasks Wang et al. (2022), while abstract syntax tree encodings integrated with Transformers improve code reasoning Hellendoorn et al. (2020). More recently, graph-guided prompting has emerged, where GNNs extract relational substructures (e.g., multi-hop paths, neighborhoods) and feed them to LLMs as structured prompts, enabling multi-hop reasoning without modifying the architecture Chen et al. (2024b).

Challenges remain in *scalability*, due to high communication and memory costs in large graphs Shao et al. (2022); Huang et al. (2024); Liu et al. (2023); in *semantic alignment*, bridging graph and token embeddings Wang et al. (2024); Li et al. (2023b); and in *evaluation*, with new benchmarks such as GLBench and LLMNodeBed beginning to standardize comparisons Zhou et al. (2024); Chen et al. (2025).

2.4 CROSS-ATTENTION FOR GRAPH LEARNING

Cross-attention is used for reasoning that jointly leverages symbolic graph structure and natural language priors, with promising applications in knowledge graph completion (Yao et al., 2022), scientific discovery (Yang et al., 2023), and multi-hop reasoning tasks (Chen et al., 2023). Ongoing challenges include balancing modality dominance, designing scalable cross-attention schemes for large graphs, and ensuring semantic alignment between embedding spaces.

3 METHOD

In this section, we present GINAT. Our approach extends the classical message-passing paradigm in graph learning. Message-passing layers are conventionally divided into three abstract components: *message*, *aggregate*, and *update* (see Appendix F for more information). In GINAT, we redefine the message computation step by incorporating semantic information derived from a pre-trained, but not finetuned, LLM. Each graph is associated with a prompt that describes *the nature* of the graph, that is, what information it represents. The prompt is processed by an LLM, and the output embeddings of its last layer are employed to enrich the original messages before aggregation, thereby introducing contextualized knowledge in the graph learning process. This idea is illustrated in Figure 1, and we provide examples for the prompts used in Figure 3.

GINAT: Formal Introduction. Let us consider a message-passing layer defined by the $message(\cdot)$, $aggregate(\cdot)$, $update(\cdot)$ functions. In GINAT, we make no changes to those methods, but we change the data flow from the $message(\cdot)$ to $aggregate(\cdot)$. Specifically, let $G = (V, E)$ be a graph with $|V|$ nodes and $|E|$ edges, with $\mathbf{H}^{(l)} \in \mathbb{R}^{|V| \times d_{l-1}}$ being node embeddings at layer l . Finally, we denote by $\mathcal{H}_T \in \mathbb{R}^{L \times d_T}$ the LLM-generated embedding of the prompt associated with the graph, and a learnable linear projection matrix $\mathbf{W} \in \mathbb{R}^{d_l \times d_T}$. The output $\mathbf{H}^{(l+1)}$, that is, the augmented message-passing layer, is then obtained via the following sequence of steps:

$$\mathbb{R}^{|L| \times d_l} \ni \tilde{\mathcal{H}}_T = \text{LayerNorm}(\mathcal{H}_T) \mathbf{W}^T, \quad (1)$$

$$\mathbb{R}^{|E| \times d_l} \ni \mathbf{M}^{(l)} = \text{message}(\mathbf{H}^{(l)}, E), \quad (2)$$

$$\mathbb{R}^{|E| \times d_l} \ni \tilde{\mathbf{M}}^{(l)} = \mathbf{M}^{(l)} + \xi \cdot \text{CrossAttention}(\tilde{\mathbf{M}}^{(l)}, \tilde{\mathcal{H}}_T), \quad (3)$$

$$\mathbb{R}^{|V| \times d_l} \ni \mathbf{H}^{(l+1)} = \text{update}(\mathbf{H}^{(l)}, \text{aggregate}(\tilde{\mathbf{M}}^{(l)})). \quad (4)$$

Figure 1 illustrates this process in a more intuitive manner. The CrossAttention term allows passed messages to selectively attend over contextualized prompt embeddings \mathcal{H}_T . The resulting formulation unifies structural and semantic information, offering a principled way of integrating pretrained language representations into graph message passing. Importantly, when the cross-attention weights are set to zero, the formulation reduces to the standard MPNN, implying that our method is at least as expressive as the MPNN we extend, in the sense of the WL test.

Scaling the Cross-Attention Output. ξ in Equation (3) is a learned scaler, that serves two purposes. First, it scales the cross-attention output. This allows the model to learn the appropriate scale of the attention mechanism compared to the MPNN’s output. This allows stable training and prevents vanishing gradients due to scale mismatch between the components. Second, it provides **explainability**. The value of ξ implies how important the model finds the information carried in the prompt embedding to be. Furthermore, it allows the model to lean $\xi = 0$, implying the added information isn’t relevant to the task.

Graph Level Aggregation. For graph-level classification, GINAT employs a virtual node, connected to all nodes in the target graph. After the forward pass, this node’s output representation is treated as the graph’s latent features.

3.1 THE CONTRIBUTION OF LLMs IN GINAT

By leveraging the vast knowledge encoded in LLMs, trained on massive corpora of text data, our method goes beyond the standard setting TAGs. Instead, it provides a more general framework in which external semantic priors complement structural message passing, thereby enhancing the representational capacity of even simple GNN architectures. The effectiveness of this integration is demonstrated through empirical results presented in Section 5.

Choice of LLM. LLMs can be proprietary or open-source, and used in pre-trained, fine-tuned, or instruction-tuned configurations. In this work, we use open-source models in their pre-trained form, since the LLM serves only as an encoder, and this provides direct access to its prior knowledge,

unaffected by alignment or tuning. While fine-tuning within our framework is a promising direction, it is resource-intensive; we focus here on testing the hypothesis that textual graph descriptions encoded by an LLM enhance inference.

4 THEORETICAL PROPERTIES OF GINAT

In this section, we provide an analysis of GINAT from model design and theoretical perspectives. We begin by presenting the relevance of cross-attention to our goal, continue by formally establishing the enhanced representation capabilities of our approach, and conclude by proving that we maintain permutation equivariance.

Message passing is the dominant framework for modern GNNs, formalizing relational information exchange between nodes. At each iteration $t+1$, a node updates its representation by aggregating messages from neighbors' features at iteration t via permutation-invariant operators such as sum, mean, or attention (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018). This captures local dependencies while preserving permutation equivariance (Appendix B). Stacking multiple layers allows nodes to integrate information from larger neighborhoods, enabling GNNs to approximate diverse graph functions. However, their expressivity is limited by the 1-WL test (Xu et al., 2019), motivating extensions such as higher-order architectures (Morris et al., 2019), positional encodings (Rampášek et al., 2022; Eliasof et al., 2023), and random feature initializations Abboud et al., 2020.

We follow these trends and propose another method to extend the expressive power of an MPNN. We equip the MPNN with a cross-attention mechanism (described in detail in Appendix E), which learns to extract relevant information from global, external feature vectors (e.g., embedded prompts). This is used to enrich the messages passed by the MPNN and extend its capacity.

4.1 ENHANCED GRAPH REPRESENTATION CAPABILITIES

Our approach can successfully distinguish between graph structures that standard MPNNs fail to differentiate. It is well established that MPNNs are at most as expressive as the 1-WL isomorphism test Xu et al. (2019). However, the 1-WL test is known to be incapable of distinguishing between certain families of graphs, such as k -regular graphs (where every node has degree k). A classical example is the cycle graph C_6 and the disjoint union of two triangles ($C_3 \cup C_3$), illustrated in Figure 2. The former consists of a single 6-node cycle, while the latter consists of two disconnected 3-cycles. For the interested reader, we provide a proof for the failure of MPNNs to separate these two different graphs in Lemma 3 in Appendix A. To establish the benefits of our GINAT, we prove the following lemma in Appendix A:

Lemma 1 (GINAT Achieves Separation Given Informative Global Features). *Let G_1 be the graph comprised of two disconnected triangles and G_2 the graph containing a 6-cycle. Let both graphs be initialized such that every node has an identical feature vector x . With each graph, we associate a global feature vector g_1, g_2 , respectively. Redefine the message sent to each node during message passing as*

$$m_{vu}^{(l)} = \text{message} \left(h_u^{(l-1)} \right), u \in N(v), \quad (5)$$

$$m'_{vu}^{(l)} = m_{vu}^{(l)} + \text{CrossAttention} \left(m_{vu}^{(l)}, g \right) \quad (6)$$

where $h_v^{(l)} = \text{update} \left(h_v^{(l-1)}, \text{aggregate} \left(\left\{ \left(\text{message} \left(h_u^{(l-1)} \right) \mid u \in N(v) \right\} \right) \right) \right)$. For a 2-layer MPNN, there exists a choice of $g_1 \neq g_2$ such that the final readout of both graphs is distinct, $H_{G_1} \neq H_{G_2}$, and there exists a choice of $g_3 \neq g_4$ (which are not linearly dependent) such that the final readout of both graphs is identical, $H_{G_1} = H_{G_2}$.

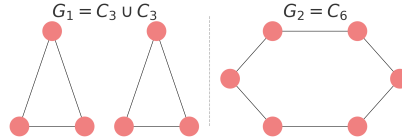


Figure 2: Illustrations of G_1, G_2 as described in Section 4.1. These non-isomorphic graphs are indistinguishable by standard MPNNs.

```
[ "Molecular structure where atoms are nodes and
edges are their connecting bonds.",
"A graph where nodes are atoms and edges show their
covalent connections.",
"Each node signifies an atom, and each edge
signifies a chemical bond.", ... ]
```

Figure 3: Examples of prompts, used to describe molecular graphs, such as OGBG – MOLHIV and ZINC. The entire prompts dataset is provide in Appendix H.

Lemma 1 establishes that an MPNN equipped with GINAT can separate G_1 from G_2 given extra feature vectors that provide *relevant information* on the graphs, or a specific task. Furthermore, it follows from the Lemma proof that when the given extra feature vectors are *not informative*, our approach will not separate the graphs. This is a *desired property*: augmenting two graphs with different feature vectors in order to obtain distinct representations is trivial, while GINAT has the ability to augment the input graph data in an informed and adept way.

On the Benefit of LLM-embedded Prompts. LLMs provide the ability to meaningfully represent text, while also using the rich prior they learn during pretraining. This could yield a representation of text that not only captures the meaning, but also provides context and additional information not present in the raw text. This capability makes LLMs the perfect candidates to generate informative extra features, as the results in Section 5 show.

Permutation Equivariance in GINAT. Permutation equivariance is a core property of GNNs and graph learning techniques. While it is well-established that cross-attention is permutation equivariant, we provide a concrete proof for this statement in Proposition 5. Proposition 2 establishes that our approach maintains permutation equivariance of MPNNs with a complete proof given in Appendix B:

Proposition 2 (GINAT Preserves Permutation Equivariance). *Standard GNNs are permutation equivariant: permuting the input node order results in the same permutation applied to the output embeddings. This property is preserved if the message function incorporates cross-attention.*

Formally, consider a message passing layer of the form

$$\tilde{h}_v^{(t+1)} = \text{update} \left(h_v^{(t)}, \text{aggregate} \left(\{m_{uv}^{(t)} : u \in \mathcal{N}(v)\} \right), \text{CrossAttention} \left(h_v^{(t)}, H_T \right) \right), \quad (7)$$

where $h_v^{(t)}$ is the embedding of node v at layer t , $\mathcal{N}(v)$ denotes its neighborhood, $m_{uv}^{(t)}$ is the message from node u to v , update is a permutation-invariant aggregation function, CrossAttention is a cross-attention operator as defined in Appendix E with $h_v^{(t)}$ as query and H_T as keys/values, and update is the node-wise update function. Then the resulting GNN layer is permutation equivariant with respect to the nodes of the input graph.

5 EXPERIMENTS

To demonstrate the utility of GINAT and better understand its impact, we designed experiments structured around the following guiding questions:

- (Q1) Can GINAT provide evidence that our core hypothesis is true, and GNNs can benefit from a global description of the processed graph?
- (Q2) Are the gains observed with GINAT attributed to the informative embeddings produced by the LLM, rather than to indirect computational effects?
- (Q3) How does GINAT contribute to enhancing the overall capability of GNNs?

Table 1: Graph Performance on Molecular Datasets. Some of the datasets contain edge features; We used those features in GAT, GATv2, GIN[E] that support them by definition. GINAT results are highlighted in red. Best results on for each GNN-dataset pair are in **bold**.

Model ↓ Metric →	MUTAG Acc. ↑	ENZYMES Acc. ↑	PROTEINS Acc. ↑	NCI1 Acc. ↑	NCI109 Acc. ↑	MOLHIV ROC-AUC ↑	ZINC MAE ↓
GCN	0.633 ±0.17	0.344 ±0.10	0.685 ±0.02	0.737 ±0.02	0.748 ±0.03	0.769 ±0.01	0.392 ±0.01
	0.733 ±0.12	0.533 ±0.04	0.744 ±0.03	0.763 ±0.02	0.791 ±0.01	0.785 ±0.02	0.391 ±0.08
GraphSAGE	0.667 ±0.10	0.250 ±0.02	0.732 ±0.02	0.750 ±0.03	0.750 ±0.01	0.728 ±0.01	0.517 ±0.02
	0.733 ±0.10	0.633 ±0.00	0.759 ±0.02	0.763 ±0.03	0.791 ±0.01	0.770 ±0.02	0.502 ±0.00
GAT	0.667 ±0.08	0.394 ±0.03	0.670 ±0.04	0.702 ±0.07	0.683 ±0.05	0.721 ±0.02	0.402 ±0.02
	0.767 ±0.06	0.506 ±0.08	0.720 ±0.04	0.783 ±0.01	0.775 ±0.02	0.748 ±0.01	0.344 ±0.02
GATv2	0.667 ±0.10	0.339 ±0.05	0.673 ±0.03	0.738 ±0.02	0.718 ±0.02	0.712 ±0.01	0.359 ±0.01
	0.783 ±0.05	0.500 ±0.04	0.729 ±0.06	0.770 ±0.02	0.791 ±0.02	0.751 ±0.01	0.315 ±0.01
GIN[E]	0.650 ±0.04	0.672 ±0.03	0.747 ±0.03	0.807 ±0.03	0.821 ±0.01	0.753 ±0.01	0.233 ±0.01
	0.700 ±0.10	0.683 ±0.04	0.753 ±0.06	0.817 ±0.03	0.833 ±0.01	0.784 ±0.00	0.136 ±0.00
GraphConv	0.667 ±0.09	0.517 ±0.17	0.723 ±0.04	0.806 ±0.02	0.794 ±0.00	0.768 ±0.00	0.225 ±0.02
	0.733 ±0.15	0.594 ±0.03	0.744 ±0.04	0.817 ±0.03	0.816 ±0.01	0.788 ±0.01	0.197 ±0.00
GraphGPS	0.783 ±0.09	0.644 ±0.15	0.482 ±0.10	0.799 ±0.03	0.808 ±0.01	0.752 ±0.02	0.216 ±0.06
	0.817 ±0.09	0.694 ±0.03	0.765 ±0.04	0.805 ±0.04	0.819 ±0.01	0.777 ±0.02	0.186 ±0.01

5.1 EXPERIMENTAL SETUP

In our experiments, we integrate GINAT into widely used GNN baselines (Tables 1, 2). The LLM is frozen and used only once to embed prompts, producing fixed representations for downstream tasks. Results are averaged over three runs with distinct, predefined seeds, and for each run we trained GINAT for 100–300 epochs, reporting the test score from the best validation checkpoint. All baseline results are from our own runs under identical splits and configurations to ensure fairness. Pretrained LLMs were obtained from HuggingFace, and we use backbone architectures PyTorch Geometric implementations (Fey & Lenssen, 2019). In all experiments, we initialized the cross-attention scaler $\xi = 0.1$ (see Section 3 for more details).

Models’ Parameters Count. Since GINAT utilizes cross-attention with LLM embeddings, whose feature dimensions are on the order of 10^3 , the layers in GINAT are larger than those in the GNN baselines typically reported in graph learning literature. To maintain a fair comparison, and to rule out the possibility that improvements arise merely from increased capacity and larger parameter count, we assign GNN baselines the same parameter budget as GINAT. Specifically, we retain architectural and configurational similarity by evaluating baselines with the same number of layers as GINAT, while adjusting the hidden dimension so that their parameter counts match as closely as possible.

Graph-Describing Prompts Data. We pair each graph dataset with a prompt dataset of approximately 200 prompts. As outlined in Section 1, these prompts describe the nature of the graph’s data modality. This allows us to use the same prompt dataset for different datasets and tasks from the same data modality. For example, we use the same molecular graph-describing prompts for both OGBG – MOLHIV and ZINC. We split the prompt dataset randomly in an 8-1-1 ratio. In a training epoch, each graph sample is paired with a single prompt embedding drawn at random from the training split. For validation and test evaluation, prompts are drawn in fixed order from the corresponding splits. See Appendix H for listings of the used prompts.

A Remark on Results. Owing to resource constraints, we did not perform exhaustive hyperparameter tuning. Each dataset-method pair was tested under three layer depths and two widths, while several parameters (e.g., learning rate, scheduling, embedding dimensions, or different normalization layers) were left at default values. This may explain deviations between our baseline results and those published elsewhere. Nevertheless, our goal is to establish the relative advantage of GINAT.

Table 2: Social-network datasets accuracy (higher is better). GINAT results are highlighted in red. Best results on for each GNN-dataset pair are in **bold**. These graphs datasets are featureless; Baselines used random node features, while GINAT used constant node features.

	REDDIT-BINARY	REDDIT-MULTI5K	REDDIT-MULTI12K	IMDB-BINARY	IMDB-MULTI
GCN	0.785 \pm 0.06 0.887 \pm 0.01	0.512 \pm 0.04 0.554 \pm 0.04	0.456 \pm 0.01 0.477 \pm 0.00	0.607 \pm 0.04 0.747 \pm 0.04	0.371 \pm 0.01 0.489 \pm 0.02
GAT	0.510 \pm 0.06 0.930 \pm 0.01	0.218 \pm 0.02 0.552 \pm 0.02	0.214 \pm 0.01 0.493 \pm 0.01	0.507 \pm 0.01 0.753 \pm 0.05	0.336 \pm 0.03 0.478 \pm 0.02
GATv2	0.510 \pm 0.06 0.928 \pm 0.02	0.198 \pm 0.02 0.572 \pm 0.03	0.214 \pm 0.01 0.496 \pm 0.00	0.500 \pm 0.00 0.743 \pm 0.04	0.327 \pm 0.03 0.482 \pm 0.03
GIN[E]	0.935 \pm 0.02 0.945 \pm 0.01	0.550 \pm 0.03 0.552 \pm 0.03	0.498 \pm 0.01 0.504 \pm 0.01	0.723 \pm 0.02 0.740 \pm 0.02	0.422 \pm 0.04 0.482 \pm 0.04
GraphConv	0.917 \pm 0.01 0.927 \pm 0.03	0.570 \pm 0.02 0.572 \pm 0.03	0.489 \pm 0.01 0.498 \pm 0.01	0.737 \pm 0.03 0.753 \pm 0.05	0.466 \pm 0.05 0.491 \pm 0.03
GraphGPS	0.777 \pm 0.04 0.928 \pm 0.01	0.339 \pm 0.10 0.510 \pm 0.01	0.218 \pm 0.01 0.477 \pm 0.03	0.737 \pm 0.03 0.750 \pm 0.04	0.416 \pm 0.05 0.442 \pm 0.04

5.2 GINAT IMPROVES GRAPH INFERENCE PERFORMANCE

In our main experiment, we seek to address (Q1) and establish the contribution of augmenting graph data with global, embedded textual descriptions of the graphs using GINAT.

Settings. We compare the performance of 7 well-known GNN baselines with their augmented counterparts on several graph-level datasets (an overview of the evaluated datasets is given in Appendix C). Specifically, we used standard MPNNs: GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), GATv2 (Brody et al., 2022), GraphConv (Morris et al., 2019), GraphSage (Hamilton et al., 2017), and GIN (Xu et al., 2019) or GINE (Hu et al., 2020) when edge features were available. We also use the graph transformer GraphGPS (Rampášek et al., 2022) as a baseline. In these experiments, we used Qwen3-4B as the embedding LLM.

Results. The performance of GINAT against the baselines, shown in tables 1, 2 establish the effectiveness of GINAT across dataset sizes, tasks (binary and multiclass classification, regression), graph sizes and sparseness, and data domains. The configurations used for each dataset are listed in Appendix D.

5.3 ABLATION STUDY: LLMs PROVIDE INFORMATIVE FEATURES

While our main experiment established that GINAT delivers performance gains, these improvements may not necessarily result from the inclusion of global features alone; factors such as injected randomness could also contribute to downstream performance. To investigate this, we tested GINAT with different LLMs embedding using the same prompts on an identical dataset and GNN configuration. Crucially, we also conducted a controlled experiment where the LLM embeddings were replaced by random Gaussian vectors of comparable size.

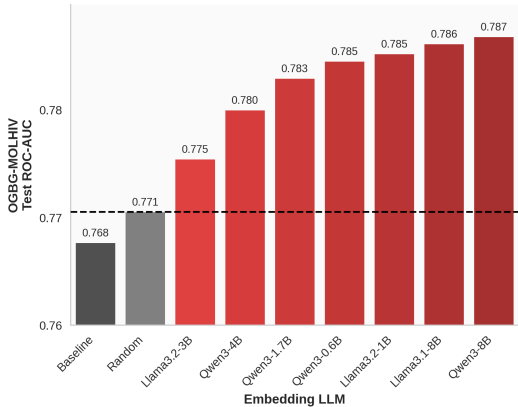


Figure 4: Performance of GINAT on MOLHIV with a GCN backbone, using various LLMs (all 5 layers, hidden size 512). All LLMs outperform the baseline and the random-noise control, marked by the black line. Larger models achieve the best results, supporting the claim that gains stem from meaningful prompt information.

In this experiment, we train GINAT on the OGBG-MOLHIV dataset using GCN backbone. We evaluate GINAT using several LLMs: the 0.6B, 1.7B, 4B, and 8B versions of Qwen3 (Yang et al., 2025), and Llama3.1-8B, Llama3.2-3B, and Llama3.2-1B (Grattafiori et al., 2024). All models were configured with 5 layers and a hidden dimension of 512. The results are illustrated in Figure 4, indicating that all tested LLMs yield substantial improvements over both the baseline and the random-noise controlled experiment. While the performance margin between models is not large, the strongest results are achieved with larger LLMs. This result provides further evidence that the gains stem from the information in the prompts - best captured by more expressive LLMs. Moreover, we see that LLMs consistently improve baseline performance.

5.4 NODE CLASSIFICATION: WHEN DO WE BENEFIT FROM GINAT?

Having established, through the first two experiments in Sections 5.2–5.3, both the effectiveness of LLM-embedded prompts and the capacity gains of GINAT, we next investigate its underlying properties. In particular, we seek to answer (Q3): is the contribution of GINAT *local* or *global*?

The answer is nontrivial: external features provide a global view of the graph, yet the embedded prompts interact locally with the GNN messages, rather than only after a *readout* aggregation.

To shed light on this matter, we apply GINAT to two large-scale node-classification benchmarks: OGBN-ARXIV and OGBN-PRODUCTS, each accompanied by a suitable prompt dataset, as discussed in Appendix H. Below, we discuss the results, which are presented in Figure 5.

Node-Level Settings. We adopt experimental settings parallel to those of the graph-level settings. Specifically, for every batch of nodes sampled by GraphSAINT (Zeng et al., 2020), we draw a matching batch of prompts, and treat the sampled subgraph as a graph as in the graph-level settings.

GINAT’s Contribution is Global. From Figure 5, it is evident that GINAT does not yield comparable gains on node-level tasks. This aligns with findings in the literature, which show that such tasks rely mainly on local neighborhood features rather than global information (Ying et al., 2019; Gasteiger et al., 2019; Wu et al., 2019; Hamilton et al., 2017). We therefore conclude that GINAT *supplements GNNs globally*. For additional insights and analysis of node-level results, we refer the reader to Appendix G.

6 CONCLUSION

We introduced GINAT — a novel approach showing that graph representation learning can benefit from global graph descriptions, embedded and enriched by an LLM. GINAT leverages a cross-attention mechanism to integrate the two modalities and provably increases GNN capacity while maintaining permutation equivariance. Through extensive experiments across multiple domains and benchmarks, supported by newly constructed prompt datasets, we provided empirical evidence for the effectiveness of the approach. GINAT consistently improves performance on graph-level tasks, achieving gains around 10% improvement over strong baselines, while operating under the same parameter budget.

The theoretical and empirical gains in this work lay the foundation for numerous extensions and improvements, including alternative conjoining mechanisms and diverse prompting strategies.

Limitations. Currently, the improvements of our method are most pronounced in graph-level tasks, a consequence of the global nature of the prompts. We view extending the framework in this direction as an exciting opportunity for future research.

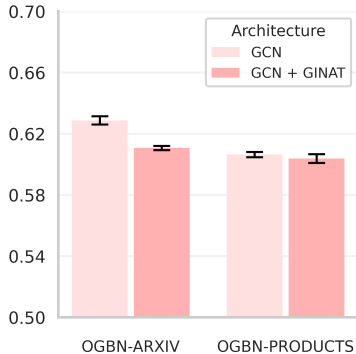


Figure 5: Performance of GINAT on node-classification benchmarks OGBN-ARXIV and OGBN-PRODUCTS. The results show that GINAT’s contributes on a *global* level.

486 AUTHORS' STATEMENTS
487

488 **Ethics Statements.** This work does not involve human subjects, personal or sensitive data, or
489 applications in high-risk domains, to the best of our knowledge. All datasets used in our experiments
490 are publicly available benchmark graph datasets (e.g., OGB) and were used in compliance with
491 their respective licenses. The large language models employed in our study are publicly released
492 open-source models obtained through HuggingFace. Our method is designed as a general framework
493 for improving graph learning and does not target or promote harmful applications. We acknowledge
494 that, as with any machine learning system, fairness and bias issues may arise if the approach is applied
495 to sensitive data; however, such applications are beyond the scope of this paper.

496 **Reproducibility Statement.** We have taken care to ensure the reproducibility of our results.
497 Complete details of datasets, model architectures, training settings, and hyperparameters are provided
498 in the main text and Appendix. The prompt datasets used are documented in Appendix H. All baseline
499 implementations follow the official PyTorch Geometric library (Fey & Lenssen, 2019), and pretrained
500 LLMs are obtained from HuggingFace repositories. All experiments used 630, 1312, 15052010 as
501 seeds. Code for GINAT, including training scripts and evaluation pipelines, is available in the
502 anonymised git repository <https://anonymous.4open.science/r/llm4gfm-3FBE/>
503

504 **Usage of Large Language Models in This Work.** LLMs were used in this work for several
505 supportive purposes, including coding assistance, text editing, grammar refinement, and LaTeX
506 formatting. In particular, they were employed to improve the clarity and coherence of the proofs
507 accompanying our theoretical claims, and to expand the prompts datasets. Their use allowed the
508 authors to focus more directly on research contributions rather than on ancillary editing tasks.
509

510 REFERENCES

- 511 Ralph Abboud, .Ismail .Ilkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising power
512 of graph neural networks with random node initialization. In *International Joint Conference on*
513 *Artificial Intelligence*, 2020. URL <https://api.semanticscholar.org/CorpusID:222134198>.
514
- 515 Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel
516 Lenc, Arthur Mensch, Malcolm Reynolds, Eliza Rutherford, et al. Flamingo: a visual language
517 model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*,
518 2022.
519
- 520 Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal,
521 Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation Models Defining a
522 New Era in Vision: A Survey and Outlook. *IEEE Transactions on Pattern Analysis & Machine*
523 *Intelligence*, 47(04):2245–2264, April 2025. ISSN 1939-3539. doi: 10.1109/TPAMI.2024.
524 3506283. URL [https://doi.ieeecomputersociety.org/10.1109/TPAMI.2024.](https://doi.ieeecomputersociety.org/10.1109/TPAMI.2024.3506283)
525 3506283.
- 526 Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International*
527 *Conference on Learning Representations*, 2022. URL [https://openreview.net/](https://openreview.net/forum?id=F72ximsx7C1)
528 [forum?id=F72ximsx7C1](https://openreview.net/forum?id=F72ximsx7C1).
529
- 530 Haibo Chen, Xin Wang, Zeyang Zhang, Haoyang Li, Weigao Wen, Ling Feng, and Wenwu Zhu.
531 Curriculum gnn-llm alignment for text-attributed graphs. In *ICLR 2025 (under review)*, 2024a.
532 Curriculum-based alignment to balance textual and structural learning difficulties in TAGs.
- 533 Hao Chen, Tianyi Sun, Wayne Xin Zhao, and Ji-Rong Wen. Llm-based node classification algorithms:
534 Benchmarking & testbed. *arXiv preprint arXiv:2502.00829*, 2025.
- 535 Yuxiao Chen, Yizhou Zhang, Yujia Zhao, Jie Li, and Yizhou Sun. Kg-llm: Linking knowledge graphs
536 with large language models for reasoning. *arXiv preprint arXiv:2310.08995*, 2023.
537
- 538 Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei
539 Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (llms)
in learning on graphs. *SIGKDD Explor. Newsl.*, pp. 42–61, March 2024b.

- 540 Moshe Eliasof, Fabrizio Frasca, Beatrice Bevilacqua, Eran Treister, Gal Chechik, and Haggai
541 Maron. Graph Positional Encoding via Random Feature Propagation. In *Proceedings of the 40th*
542 *International Conference on Machine Learning*, 2023.
- 543 Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric.
544 In *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL
545 <https://arxiv.org/abs/1903.02428>.
- 546 Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate:
547 Graph neural networks meet personalized pagerank. In *International Conference on Learning*
548 *Representations (ICLR)*, 2019.
- 549 Dedre Gentner and Jennifer Asmuth. Metaphoric extension, relational categories, and abstraction.
550 *Language, Cognition and Neuroscience*, 34(10):1298–1307, 2019. doi: 10.1080/23273798.2017.
551 1410560. URL <https://doi.org/10.1080/23273798.2017.1410560>.
- 552 Aaron Grattafiori, Laurens van der Maaten, Alexandre Defossez, and et al. The llama 3 herd
553 of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL
554 <https://arxiv.org/abs/2407.21783>.
- 555 Yiming Gu, Jie Yang, Wen Zhang, Yixin Li, and Huajun Wang. Empowering graph neural
556 network-based computational drug repositioning with llm-enhanced embeddings. *arXiv preprint*
557 *arXiv:2404.01234*, 2024. LLM-derived embeddings used as enhanced node features in heteroge-
558 neous GNN for drug–disease prediction.
- 559 William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs.
560 In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- 561 Vincent J Hellendoorn, Christian Bird, and Earl T Barr. Global relational models of source code. In
562 *ICLR 2020*, 2020.
- 563 Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang.
564 Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD*
565 *Conference on Knowledge Discovery and Data Mining*, pp. 594–604, 2022.
- 566 Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec.
567 Strategies for pre-training graph neural networks. In *International Conference on Learning Repre-*
568 *sentations (ICLR)*, 2020. URL <https://openreview.net/forum?id=HJlxPhHFvB>.
- 569 Xu Huang, Juncheng Wu, Jiawei Li, and Yizhou Zhang. Catgcn: Cost-efficient and scalable
570 distributed training for graph neural networks. *arXiv preprint arXiv:2404.02300*, 2024.
- 571 John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction
572 with alphafold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-2.
- 573 Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks.
574 *arXiv preprint arXiv:1609.02907*, 2017.
- 575 Kapa Korta and John Perry. Pragmatics. In Edward N. Zalta and Uri Nodelman (eds.), *The Stanford*
576 *Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2024 edition,
577 2024.
- 578 Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-
579 image pre-training with frozen image encoders and large language models. *arXiv preprint*
580 *arXiv:2301.12597*, 2023a.
- 581 Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A
582 survey of graph meets large language model: Progress and future directions. In *arXiv preprint*
583 *arXiv:2311.12399*, 2023b. Presents LLM roles as enhancer, predictor, and alignment component
584 in graph tasks.
- 585 Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan
586 Zhang. One for all: Towards training one graph model for all classification tasks. In *The Twelfth*
587 *International Conference on Learning Representations*, 2024. URL [https://openreview.](https://openreview.net/forum?id=4IT2pgc9v6)
588 [net/forum?id=4IT2pgc9v6](https://openreview.net/forum?id=4IT2pgc9v6).

- 594 Yang Liu, Yifan Zhao, Ziqi Wang, and Jie Tang. Efficient large language models fine-tuning on
595 graphs. *arXiv preprint arXiv:2312.04737*, 2023.
596
- 597 Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav
598 Rattan, and Martin Grohe. Weisfeiler and leman go neural: higher-order graph neural networks.
599 In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First
600 Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on
601 Educational Advances in Artificial Intelligence*, 2019.
- 602 Abdul-Aziz Mumuni and Fuseini Mumuni. Large language models for artificial general intelligence
603 (AGI): A survey of foundational principles and approaches. *arXiv preprint arXiv:2501.03151*,
604 2025. URL <https://arxiv.org/abs/2501.03151>.
605
- 606 Chengwei Qin, Yeyun Chen, Yi Feng, et al. Large language models meet NLP: A survey. *arXiv
607 preprint arXiv:2405.04646*, 2024. URL <https://arxiv.org/abs/2405.04646>.
- 608 Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Do-
609 minique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Proceedings of the
610 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook,
611 NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.
612
- 613 Yuke Shao, Lei Chen, et al. Distributed graph neural network training: A survey. *arXiv preprint
614 arXiv:2211.00216*, 2022.
- 615 Yu Tian et al. Graph neural prompting with large language models. In *AAAI 2024*, 2024. Graph
616 Neural Prompting (GNP): embeds KG knowledge into LLMs via GNN-derived prompts.
617
- 618 Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua
619 Bengio. Graph attention networks. In *International Conference on Learning Representations
620 (ICLR)*, 2018.
- 621 Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pas-
622 cal Frossard. Digress: Discrete denoising diffusion for graph generation. In *Proceedings
623 of the Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL
624 <https://openreview.net/forum?id=UaAD-Nu86WX>.
625
- 626 Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolu-
627 tional networks for recommender systems. In *Proceedings of the AAAI Conference on Artificial
628 Intelligence (AAAI)*, 2019.
- 629 Xiaoyu Wang, Le Wu, Jian Gao, and Enhong Chen. Molecular graph-enhanced language models
630 for property prediction and molecule generation. In *Proceedings of the 36th AAAI Conference on
631 Artificial Intelligence (AAAI)*, 2022.
632
- 633 Zhenyu Wang, Kai Li, Wayne Xin Zhao, and Ji-Rong Wen. Llms as zero-shot graph learners:
634 Alignment of gnn representations with llm token embeddings. In *NeurIPS 2024*, 2024.
- 635 Jamie Williams, Noah Russell, and Derek Irwin. On the notion of abstraction in systemic functional
636 linguistics. *Functional Linguistics*, 4(1):13, 2017.
637
- 638 Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger.
639 Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov
640 (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of
641 *Proceedings of Machine Learning Research*, pp. 6861–6871. PMLR, 09–15 Jun 2019. URL
642 <https://proceedings.mlr.press/v97/wu19e.html>.
- 643 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
644 networks? In *International Conference on Learning Representations (ICLR)*, 2019.
645
- 646 Hao Yan, Jialin Chen, Jiasheng Zhang, Xinyan Chen, Mengting Tang, and Rex Ying. A comprehensive
647 study on text-attributed graphs: Benchmarking and rethinking. In *Advances in Neural Information
Processing Systems*, volume 36, 2023a.

- 648 Hao Yan, Jialin Chen, Jiasheng Zhang, Mengting Tang, Xinyan Chen, and Rex Ying. Learning
649 on large-scale text-attributed graphs via variational inference. In *International Conference on*
650 *Learning Representations*, 2023b.
- 651
652 Hao Yan, Yu Li, Jiasheng Zhang, Mengting Tang, Xinyan Chen, Fei Wang, and Rex Ying. Harnessing
653 explanations: Llm-to-llm interpreter for enhanced text-attributed graph representation learning. In
654 *12th International Conference on Learning Representations, ICLR 2024*, 2024.
- 655 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and et al. Qwen3 technical
656 report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL <https://arxiv.org/abs/2505.09388>.
- 657
658 Chao Yang, Yao Xu, Zhilin Wang, Tuo Zhao, et al. Graphllm: Enhancing large language models with
659 graph reasoning. *arXiv preprint arXiv:2308.08614*, 2023.
- 660
661 Quanming Yao, Shirui Pan, Yangqiu Zhang, et al. Nlp2graph: Multi-modal knowledge graph
662 construction. In *Proceedings of the 31st ACM International Conference on Information and*
663 *Knowledge Management (CIKM)*, 2022.
- 664
665 Michihiro Yasunaga, Hongyu Ren, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language
666 models with document links. In *Proceedings of the 60th Annual Meeting of the Association for*
667 *Computational Linguistics (ACL)*, 2022.
- 668
669 Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer:
670 Generating explanations for graph neural networks. In *Advances in Neural Information Processing*
671 *Systems (NeurIPS)* 32, pp. 9240–9251, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html>.
- 672
673 Yuxin You, Zhen Liu, Xiangchao Wen, Yongtao Zhang, and Wei Ai. Large language models meet
674 graph neural networks: A perspective of graph mining. *arXiv preprint arXiv:2412.19211*, 2024.
675 Taxonomy of GNN-driving-LLM, LLM-driving-GNN, and co-driving schemes.
- 676
677 Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor K. Prasanna.
678 GraphSAINT: Graph sampling based inductive learning method. In *International Conference*
679 *on Learning Representations (ICLR)*, 2020. URL <https://openreview.net/forum?id=BJe8pkHFwB>.
- 680
681 Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying.
682 Dtg: A comprehensive benchmark for dynamic text-attributed graphs. In *Advances in Neural*
683 *Information Processing Systems*, volume 37, 2024.
- 684
685 Wei Zhang, Xu Chen, and Jie Tang. Graph-augmented language models for knowledge-intensive nlp
686 tasks. *arXiv preprint arXiv:2304.04712*, 2023.
- 687
688 Jianan Zhao, Hesham Mostafa, Mikhail Galkin, Michael M. Bronstein, Zhaocheng Zhu, and Jian
689 Tang. Fully-inductive node classification on arbitrary graphs. In *International Conference on*
690 *Learning Representations*, 2024.
- 691
692
693
694
695
696
697
698
699
700
701

A EXPRESSIVENESS OF OUR APPROACH

In this appendix, we present a concrete example of two graphs indistinguishable by an MPNN that our approach successfully distinguishes. First, we prove that an MPNN, regardless of implementation, fails to distinguish these graphs. Then we show that our approach successfully distinguishes between them.

A.1 MPNNs CANNOT DISTINGUISH BETWEEN TWO TRIANGLES AND AN HEXAGON

Lemma 3. *Let G_1 be the graph comprised of two disconnected triangles and G_2 the graph containing a 6-cycle. Let both graphs be initialized such that every node has the identical feature vector x . For any number of MPNN layers with any learned weights, and for any deterministic readout function R , the resulting graph embeddings will be identical: $H_{G_1} = H_{G_2}$*

Proof. We will show by induction that for any layer l , all 12 nodes across both graphs have the exact same embedding vector $h_v^{(k)}$.

It follows that the multiset of final node embeddings is identical as well,

$$\mathcal{H}_{G_1} = \left\{ \left\{ h_v^{(k)} \mid v \in G_1 \right\} \right\} = \left\{ \left\{ h_v^{(k)} \mid v \in G_2 \right\} \right\} = \mathcal{H}_{G_2}$$

Then, since the readout function R is deterministic, the final embeddings hold

$$H_{G_1} = R(\mathcal{H}_{G_1}) = R(\mathcal{H}_{G_2}) = H_{G_2}$$

completing the proof.

Base Case: For $l = 0$, by definition, all nodes in both graphs are initialized with the same feature vector x .

Inductive Step: Assume that at layer $l - 1$, all nodes in both graphs have the same embedding, $\forall v \in G_1 \cup G_2, h_v^{(l-1)} = h^{(l-1)}$. The MPNN update is a function of the node’s own embedding and the aggregation of its neighbors’ embeddings:

$$h_v^{(l)} = \text{update} \left(h_v^{(l-1)}, \text{aggregate} \left(\left\{ \left(\text{message} \left(h_u^{(l-1)} \right) \mid u \in N(v) \right\} \right) \right) \right) \quad (8)$$

By definition, all nodes in G_1, G_2 have exactly two neighbors. Let v be a node in one of the graphs. By the induction assumption, v and its neighbors u_1, u_2 , all have the same embedding $h^{(l-1)}$. The aggregated message to v is then

$$m_v = \text{aggregate} \left(\{ \text{message} (h_{u_1}), \text{message} (h_{u_2}) \} \right) \quad (9)$$

Thus the new embedding $h_v^{(l)}$ is

$$h_v^{(l)} = \text{update} (h_{l-1}, m_v) \quad (10)$$

Since our choice of v was arbitrary, this holds for all nodes in both graphs. \square

A.2 PROOF OF LEMMA 1

For the reader’s convenience, we reiterate the lemma here.

Lemma 4 (GINAT Achieves Separation Given Informative Global Features). *Let G_1 be the graph comprised of two disconnected triangles and G_2 the graph containing a 6-cycle. Let both graphs be initialized such that every node has an identical feature vector x . With each graph, we associate a global feature vector g_1, g_2 , respectively. Redefine the message sent to each node during message passing as*

$$\begin{aligned} m_{vu}^{(l)} &= \text{message} \left(h_u^{(l-1)} \right), \quad u \in N(v) \\ m'_{vu}^{(l)} &= m_{vu}^{(l)} + \text{CrossAttention} \left(m_{vu}^{(l)}, g \right) \end{aligned} \quad (11)$$

756 where $h_v^{(l)} = \text{update} \left(h_v^{(l-1)}, \text{aggregate} \left(\left\{ \left(\text{message} \left(h_u^{(l-1)} \right) \mid u \in N(v) \right\} \right) \right) \right)$. For a 2-layer
 757 MPNN, there exists a choice of $g_1 \neq g_2$ such that the final readout of both graphs is distinct,
 758 $H_{G_1} \neq H_{G_2}$, and there exists a choice of $g_3 \neq g_4$ (which are not linearly dependent) such that the
 759 final readout of both graphs is identical, $H_{G_1} = H_{G_2}$.
 760

761 *Proof.* We prove each claim of the lemma constructively, and provide full calculations for the initial
 762 feature vector $x = [1, 3, 1, 2]^T$. For tractability, and without loss of generalization, we define
 763 *aggregate* as summation and *update* as a residual connection, i.e., $h_v^{(l)} = h_v^{(l-1)} + m_v^{(l)}$. Similar
 764 proofs can be constructed for any MPNN-suitable selection of these functions.
 765

766 **Proof of first claim** ($\exists g_1, g_2 : H_{G_1} \neq H_{G_2}$) Let
 767

$$768 \quad g_1 = [3, 1, 0, 0]^T, g_2 = [6, 1, 0, 0]^T$$

769 The proof proceeds by tracing the computation.
 770

771 All nodes in both graphs start with $h^{(0)} = x = [1, 3, 1, 2]^T$. Since all nodes have degree 2, the initial
 772 aggregated message is identical for all nodes:
 773

$$774 \quad m^{(1)} = x + x = 2x = [2, 6, 2, 4]^T \quad (12)$$

775 The attention mechanism then causes a divergence. For G_1 , the dot product is
 776

$$777 \quad m^{(1)} \cdot g_1 = [2, 6, 2, 4] \cdot [3, 1, 0, 0]^T = 12 \quad (13)$$

778 The augmented message is
 779

$$780 \quad m'_{G_1} = m^{(1)} + 12g_1 = [2, 6, 2, 4]^T + [36, 12, 0, 0]^T = [38, 18, 2, 4]^T \quad (14)$$

781 For G_2 , the dot product is
 782

$$783 \quad m^{(1)} \cdot g_2 = [2, 6, 2, 4] \cdot [6, 1, 0, 0]^T = 18 \quad (15)$$

784 The augmented message is
 785

$$786 \quad m'_{G_2} = m^{(1)} + 18g_2 = [2, 6, 2, 4]^T + [108, 18, 0, 0]^T = [110, 24, 2, 4]^T \quad (16)$$

787 Since the augmented messages are different, the updated node embeddings after the first layer are
 788 also different: $h_{G_1}^{(1)} = x + m'_{G_1} = [39, 21, 3, 6]^T$, while $h_{G_2}^{(1)} = x + m'_{G_2} = [111, 27, 3, 6]^T$.
 789

790 As the node embeddings have diverged after the first layer, all subsequent computations will also
 791 differ, guaranteeing that the final multisets of node embeddings are distinct and thus the final readouts
 792 $H_{G_1} \neq H_{G_2}$.
 793

794 **Proof of second claim** ($\exists g_3, g_4 : H_{G_1} = H_{G_2}$) For brevity, we first compute the common messages
 795 that would be generated without attention:
 796

$$797 \quad m^{(1)} = 2x = [2, 6, 2, 4]^T \quad (17)$$

798 This gives
 799

$$800 \quad h^{(1)} = x + m^{(1)} = 3x = [3, 9, 3, 6]^T \quad (18)$$

801 The next message is
 802

$$803 \quad m^{(2)} = h^{(1)} + h^{(1)} = 2h^{(1)} = [6, 18, 6, 12]^T \quad (19)$$

804 Let
 805

$$806 \quad g_3 = [-3, 1, 0, 0]^T, g_4 = [-1, 0, 1, 0]^T$$

807 The proof proceeds by tracing. At layer 1, the initial message is $m^{(1)}$. The dot product for G_1 is
 808

$$809 \quad m^{(1)} \cdot g_1 = [2, 6, 2, 4] \cdot [-3, 1, 0, 0] = -6 + 6 = 0 \quad (20)$$

Similarly for G_2 ,

$$m^{(1)} \cdot g_2 = [2, 6, 2, 4] \cdot [-1, 0, 1, 0] = -2 + 2 = 0 \quad (21)$$

810 In both cases, the attention is nullified, so the augmented message is just $m^{(1)}$, and the updated
811 embedding is $h^{(1)} = [3, 9, 3, 6]^T$ for all nodes in both graphs. At layer 2, the message is $m^{(2)}$. The
812

813 dot product for G_1 is

$$814 \quad m^{(2)} \cdot g_3 = [6, 18, 6, 12] \cdot [-3, 1, 0, 0] = -18 + 18 = 0 \quad (22)$$

816 For G_2 ,

$$817 \quad m^{(2)} \cdot g_4 = [6, 18, 6, 12] \cdot [-1, 0, 1, 0] = -6 + 6 = 0 \quad (23)$$

818 Again, the attention is nullified, and the augmented message is identical for both graphs. This ensures
819 the final node embeddings are also identical. Since the final multisets of embeddings are identical,
820 the readouts must be identical, $H_{G_1} = H_{G_2}$.
821

822 \square

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

B PRESERVATION OF EQUIVARIANCE

GNNs are inherently permutation equivariant: reordering the nodes in the input graph leads to the same reordering in the output node embeddings. Analogously, multi-modal cross-attention is also permutation equivariant with respect to the queries. In this section, we prove that our method maintains permutation equivariance - a key characteristic in any GNN.

Proposition 5. *Let $Q \in \mathbb{R}^{n_q \times d}$ denote the query matrix, $K \in \mathbb{R}^{n_k \times d}$ the key matrix, and $V \in \mathbb{R}^{n_k \times d_v}$ the value matrix. The cross-attention operator is defined as*

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V,$$

where the softmax is applied row-wise. Then for any permutation matrix $P \in \{0, 1\}^{n_q \times n_q}$, it holds that

$$\text{Attn}(PQ, K, V) = P \text{Attn}(Q, K, V).$$

That is, cross-attention is permutation equivariant with respect to its queries.

Proof. Consider the attention with permuted queries:

$$\text{Attn}(PQ, K, V) = \text{softmax}\left(\frac{(PQ)K^\top}{\sqrt{d}}\right)V.$$

By associativity of matrix multiplication,

$$(PQ)K^\top = P(QK^\top).$$

Since the softmax is applied row-wise, permuting the rows before softmax is equivalent to permuting them after softmax:

$$\text{softmax}(P(QK^\top)) = P \text{softmax}(QK^\top).$$

Therefore,

$$\text{Attn}(PQ, K, V) = (P \text{softmax}(QK^\top))V = P (\text{softmax}(QK^\top)V).$$

This simplifies to

$$\text{Attn}(PQ, K, V) = P \text{Attn}(Q, K, V),$$

proving permutation equivariance with respect to queries. \square

Proposition 6. *Standard GNNs are permutation equivariant: permuting the input node order results in the same permutation applied to the output embeddings. This property is preserved if the message function incorporates cross-attention.*

Formally, consider a message passing layer of the form

$$\tilde{h}_v^{(t+1)} = \text{update}\left(h_v^{(t)}, \text{aggregate}(\{m_{uv}^{(t)} : u \in \mathcal{N}(v)\}), \text{CrossAttention}(h_v^{(t)}, H_T, H_T)\right),$$

where $h_v^{(t)}$ is the embedding of node v at layer t , $\mathcal{N}(v)$ denotes its neighborhood, $m_{uv}^{(t)}$ is the message from node u to v , AGG is a permutation-invariant aggregation function, CrossAttn is a cross-attention operator with $h_v^{(t)}$ as query and H_T as keys/values, and U is the node-wise update function. Then the resulting GNN layer is permutation equivariant with respect to the nodes of the input graph.

Proof. Let P be a permutation matrix applied to reorder the nodes in the input graph. Permutation equivariance requires that

$$f(PX, PAP^\top) = Pf(X, A),$$

where X is the node feature matrix and A the adjacency matrix.

Consider the message-passing layer. Each node embedding $h_v^{(t)}$ acts as a query to the cross-attention module, while H_T provides the keys/values.

By Proposition 3, cross-attention is permutation equivariant with respect to its queries and permutation invariant with respect to the keys/values. Thus, applying P permutes the queries consistently: $\text{CrossAttention}(Ph_v^{(t)}, H_T) = P \text{CrossAttention}(h_v^{(t)}, H_T)$. The aggregated messages from neighbors are permutation equivariant due to the permutation-invariance of *aggretate*, i.e., $\text{aggregate}(\{m_{p(u)p(v)}^{(t)} : u \in \mathcal{N}(v)\}) = P \text{aggagate}(\{m_{uv}^{(t)} : u \in \mathcal{N}(v)\})$. Finally, the node-wise update *update* preserves the permutation, since it is applied independently to each node.

Combining these, each node update under permutation P satisfies

$$\tilde{h}_{p(v)}^{(t+1)} = \text{update}\left(Ph_v^{(t)}, P \text{aggregate}(\{m_{uv}^{(t)}\}), P \text{CrossAttention}(h_v^{(t)}, H_T)\right) = P\tilde{h}_v^{(t+1)}.$$

Hence, the GNN layer with cross-attention in the message stage is permutation equivariant. \square

C DATASETS INFORMATION

C.1 MOLECULAR GRAPH DATASETS

Table 3: Summary statistics of molecular datasets. Feature columns report only dimensionality. Totals are estimated as $\#\text{graphs} \times \text{average edges}$.

Dataset	Graphs	Nodes	Edges	Tot. E	Node dim	Edge dim
OGBG-MOLHIV	41,127	25.5	27.5	1,130,992	9	3
MUTAG	188	17.9	19.8	3,721	1	1
ENZYMES	600	32.6	62.1	37,284	19	–
PROTEINS	1,113	39.1	72.8	81,049	2	–
NCI1	4,110	29.9	32.3	132,753	1	–
NCI109	4,127	29.7	32.1	132,601	1	–
ZINC (full)	249,456	23.2	24.9	6,211,454	1	1

OGBG-MOLHIV. A molecular property prediction benchmark: binary classification of whether a molecule inhibits HIV replication. Nodes are atoms with 9 categorical features (atomic number, chirality, aromaticity, etc.); edges are bonds with 3 categorical features (bond type, stereo, conjugation).

MUTAG. A dataset of nitroaromatic compounds labeled by mutagenicity. Nodes are atoms (atom type); edges are bonds (bond type).

ENZYMES. Contains 600 protein tertiary-structure graphs classified into 6 enzyme commission (EC) classes. Nodes are secondary structure elements (SSEs) with one categorical type (helix/sheet/turn) and 18 continuous attributes. Together, this yields 19 node dimensions. Edges connect interacting SSEs.

PROTEINS. Protein tertiary-structure graphs with binary labels (enzyme vs. non-enzyme). Nodes are SSEs with a categorical type and one continuous attribute (2 total dimensions). `PROTEINS_full` provides 29 attributes.

NCI1 / NCI109. Molecules tested for anti-cancer activity. Nodes are atoms (1 categorical type dimension); edges are unlabeled. Graph labels indicate activity against cancer cell lines.

ZINC. Large dataset of drug-like molecules from the ZINC database. Nodes are atoms (1 categorical type dimension), edges are bonds (1 categorical type dimension). Used primarily for property regression tasks.

C.2 SOCIAL NETWORK GRAPH DATASETS

Table 4: Summary statistics of molecular datasets. Feature columns report only dimensionality. Totals are estimated as $\#\text{graphs} \times \text{average edges}$.

Dataset	Graphs	Nodes	Edges	Tot. E	Node dim	Edge dim
OGBG-MOLHIV	41,127	25.5	27.5	1,130,992	9	3
MUTAG	188	17.9	19.8	3,721	1	1
ENZYMES	600	32.6	62.1	37,284	19	–
PROTEINS	1,113	39.1	72.8	81,049	2	–
NCI1	4,110	29.9	32.3	132,753	1	–
NCI109	4,127	29.7	32.1	132,601	1	–
ZINC (full)	249,456	23.2	24.9	6,211,454	1	1

1026 **REDDIT-BINARY, REDDIT-MULTI-5K, REDDIT-MULTI-12K.** Each graph represents a
1027 Reddit discussion thread. Nodes are users; edges connect users if one replied to the other. Graph
1028 labels correspond to subreddit/community categories (2, 5, and 11/12 classes, respectively).
1029

1030 **IMDB-BINARY / IMDB-MULTI.** Graphs are ego-networks of movie collaborations. Nodes are
1031 actors/actresses, and edges indicate co-appearance in a movie. Graph labels are genres: IMDB-
1032 BINARY distinguishes Action vs. Romance, while IMDB-MULTI has three genres (Comedy,
1033 Romance, Sci-Fi).
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

D DATASET TRAINING CONFIGURATION

As discussed in Section 5, for each dataset and GNN backbone pair, we searched for the optimal layer count and layer width combination. The table below details the configuration used for each dataset. We reiterate that the same configuration was used for the baseline model and the LLM augmented model.

Table 5: Configuration used for molecular graph datasets, given in a layers/hidden dimension format.

	GCN	GraphConv	GraphSAGE	GIN	GAT	GATv2	GraphGPS
OGBG-MOLHIV	5 / 256	6 / 256	6 / 512	4 / 512	5 / 512	5 / 512	4 / 512
ZINC	5 / 512	5 / 512	4 / 512	5 / 256	4 / 256	4 / 256	5 / 256
MUTAG	6 / 256	4 / 256	6 / 512	6 / 256	4 / 256	4 / 256	6 / 256
PROTEINS	6 / 512	4 / 512	5 / 512	3 / 256	3 / 512	4 / 256	3 / 512
ENZYMES	6 / 512	5 / 512	5 / 512	4 / 256	5 / 512	4 / 512	4 / 512
NCI1	3 / 512	4 / 512	3 / 512	3 / 512	5 / 512	3 / 512	5 / 256
NCI109	3 / 256	6 / 512	5 / 512	6 / 256	4 / 512	6 / 512	5 / 512

Table 6: Configuration used for social network graph datasets, given in a layers/hidden dimension format.

	GCN	GraphConv	GIN	GAT	GATv2	GraphGPS
Reddit-Binary	4 / 256	5 / 512	5 / 512	5 / 512	5 / 256	4 / 512
Reddit-Multi12k	5 / 256	5 / 512	5 / 512	5 / 512	5 / 256	3 / 512
Reddit-Multi5K	4 / 256	4 / 256	3 / 512	5 / 256	5 / 256	5 / 256
IMDB-Binary	4 / 512	4 / 512	6 / 256	5 / 256	5 / 512	4 / 256
IMDB-Multi	6 / 256	5 / 256	4 / 256	5 / 256	6 / 512	4 / 512

E CROSS-ATTENTION FOR GNN-LLM INTEGRATION

Cross-attention has emerged as a central mechanism for embedding and aligning multi-modal information, enabling models to selectively integrate representations across heterogeneous modalities. For example, they were proven effective in vision-language models (Li et al., 2023a; Alayrac et al., 2022), where textual tokens are grounded in visual features through learned cross-modal interactions.

Formally, given two node feature matrices $X_G \in \mathbb{R}^{n_g \times d_g}$, $X_T \in \mathbb{R}^{n_t \times d_t}$, the cross-attention layer is comprised of three (learned) weight matrices that project these features to a shared vector space with dimension d . Without loss of generality, we will use X_G as queries and X_T as keys and values denoted by $W_Q \in \mathbb{R}^{d_g \times d}$, $W_K \in \mathbb{R}^{d_t \times d}$, $W_V \in \mathbb{R}^{d_t \times d}$. Then, we compute

$$Q = X_G W_Q^T \quad K = X_T W_K^T \quad V = X_T W_V^T$$

which are usually termed the query, key, and value matrices, respectively. The cross-attention operator is then defined as:

$$\text{CrossAttention}(X_G, X_T) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (24)$$

In our GINAT, we explore cross-attention as an interface between GNNs and LLMs. Concretely, we consider GNNs that provide embeddings $H_G \in \mathbb{R}^{|V| \times d}$ that encode structural dependencies among nodes. Specifically, as described in Section 3, we use the messages passed between nodes as queries. As keys and values, we consider LLM-generated contextualized token embeddings $H_T \in \mathbb{R}^{L \times d}$. Cross-attention is then employed to align these spaces,

$$Z_G = \text{CrossAttention}(H_G, H_T), \quad (25)$$

which enriches the messages with semantic cues from language.

F BACKGROUND: MESSAGE PASSING NEURAL NETWORKS (MPNNS)

Let $G = (V, E)$ be a graph with node set V and (directed or undirected) edge set $E \subseteq V \times V$. Each node $v \in V$ may have a feature vector $h_v^{(0)} \in \mathbb{R}^{d_0}$ and each edge $(u, v) \in E$ may have features $e_{uv} \in \mathbb{R}^{d_e}$. MPNNs produce layer-wise node representations $\{h_v^{(l)}\}_{v \in V}$ for $l = 0, \dots, L$ by repeatedly exchanging *messages* along edges and updating local states. A single layer $l \rightarrow l+1$ is defined by three components:

$$\text{message}(\cdot), \quad \text{aggregate}(\cdot), \quad \text{update}(\cdot).$$

Message. For each incoming edge $(u, v) \in E$, node u sends a message to v :

$$m_{u \rightarrow v}^{(l)} = \text{message}(h_u^{(l)}, h_v^{(l)}, e_{uv}) \in \mathbb{R}^{d_m}. \quad (26)$$

The function *message* can be a neural network (e.g., MLP) or a parametric rule (e.g., linear map, attention score). It may depend on source features $h_u^{(l)}$, target features $h_v^{(l)}$, and edge features e_{uv} .

Aggregate. Node v aggregates all incoming messages from its neighborhood $\mathcal{N}(v) = \{u \mid (u, v) \in E\}$ using a permutation-invariant operator:

$$\bar{m}_v^{(l)} = \text{aggregate}(\{m_{u \rightarrow v}^{(l)} : u \in \mathcal{N}(v)\}). \quad (27)$$

Common choices include *sum*, *mean*, *max*, learned attention-weighted sums, or more expressive set functions. Permutation invariance ensures that reordering neighbors does not change the result.

Update. Node v updates its representation by combining its previous state with the aggregated message:

$$h_v^{(l+1)} = \text{update}(h_v^{(l)}, \bar{m}_v^{(l)}) \in \mathbb{R}^{d_{l+1}}. \quad (28)$$

Typical updates are MLPs, gated mechanisms (e.g., GRU), residual additions with normalization, or transformer-style blocks. With permutation-invariant *aggregate*, the mapping $H^{(l)} \mapsto H^{(l+1)}$ is *permutation equivariant* at the node level.

Readout for graph-level tasks. For graph-level prediction, node embeddings after L layers are pooled into a single graph representation via a permutation-invariant *readout*:

$$h_G = \text{readout}(\{h_v^{(L)} : v \in V\}), \quad \hat{y}_G = f_{\text{head}}(h_G). \quad (29)$$

Common readouts include *sum/mean/max* pooling, attention pooling, Set2Set, SortPool, or hierarchical pooling. The invariance of *readout* guarantees that \hat{y}_G is independent of node ordering.

Design choices and examples.

- **GCN-style layers.** Messages linearly transform neighbors, often with degree normalization, and *aggregate* is summation/mean; *update* applies a nonlinearity.
- **GAT-style layers.** *message* uses learned attention coefficients α_{uv} over neighbors; *aggregate* is the attention-weighted sum; *update* applies projection and nonlinearity.
- **Edge-aware MPNNs.** Edge features e_{uv} modulate messages (e.g., via edge-conditioned filters).
- **Stabilization.** Residual/skip connections, normalization (batch/layer), and dropout are commonly used to improve optimization and mitigate oversmoothing.

Properties. With permutation-invariant *aggregate* (and *readout*), MPNNs are permutation-equivariant at the node level and permutation-invariant at the graph level. Expressivity grows with depth and the choice of *message/aggregate/update*, and can be related to the Weisfeiler–Leman hierarchy under suitable conditions.

Summary. An MPNN layer computes messages on edges (26), aggregates them per node (27), and updates node states (28); a graph-level predictor applies a permutation-invariant readout (29). This decomposition provides a flexible template that subsumes many popular GNN architectures while preserving the necessary symmetry constraints of graph data.

1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295

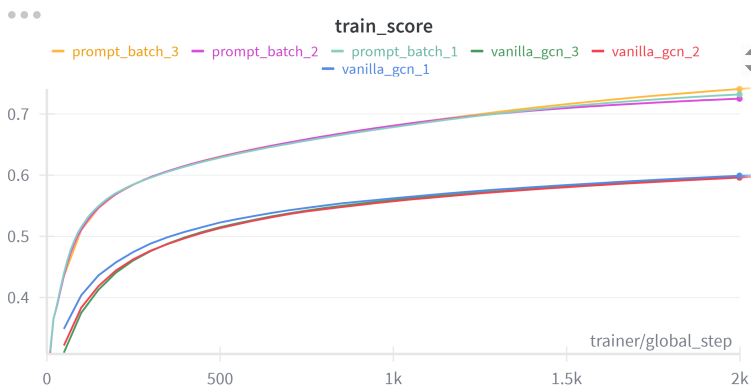


Figure 6: Batch of prompts to a single embedding for a batch of nodes configuration training fit curve for the OGBN-ARXIV dataset



Figure 7: Prompt to node configuration training fit curve for the OGBN-ARXIV dataset

G NODE CLASSIFICATION

Here, we summarize the experiments conducted to validate the claim that global descriptive prompts do not significantly contribute to the model’s generalization.

We designed two experimental setups to assess this claim. In the first approach, we employed the GraphSAINT Node Sampler to sample a batch of nodes from the graph. Analogous to the graph-level task, we also sampled a batch of prompts and extracted a single embedding representing their combined information. Cross-attention scores between these concise prompt embeddings and the node embeddings were then added as a bias to the node embeddings during the message-passing stage.

In the second approach, we adopted a prompt-per-node strategy, where each node is assigned a dedicated prompt. A cross-attention mechanism is then applied between the node and its corresponding prompt, allowing node-level interactions with the prompts.

We evaluated both approaches on the ogbn-arxiv and ogbn-products datasets and observed similar outcomes in both cases. Results, compared to a standard GCN, are shown in Figures 8. While the addition of prompts significantly improves the model’s ability to fit the training data, Figure ?? demonstrates that they provide no meaningful improvement in generalization and, in some cases, even slightly degrade performance compared to a standard GCN.

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349



Figure 8: Batch of prompts to a single embedding for a batch of nodes configuration test results for the OGBN-ARXIV dataset

1350 H PROMPTS USED IN OUR EXPERIMENTS

1351

1352

1353 In this appendix, we provide the prompts dataset we used to describe graphs in the reported experi-
1354 ments.

1355

1356 H.1 MOLECULAR GRAPHS DESCRIPTIONS DATASET

1357

1358

1359 ["A molecular graph where each node represents an atom and each
1360 edge a bond.",
1361 "Graph where atoms are nodes and the bonds between them are
1362 edges.",
1363 "This graph models a molecule with atoms as nodes and bonds as
1364 edges.",
1365 "Represents a chemical structure where nodes are atoms and edges
1366 are covalent bonds.",
1367 "Each vertex is an atom, and each edge is a chemical bond.",
1368 "A molecule's structure shown with atoms as nodes and bonds as
1369 edges.",
1370 "Nodes correspond to atoms, and edges represent the bonds
1371 connecting them.",
1372 "A graph illustrating atoms as vertices and their chemical
1373 connections as edges.",
1374 "Chemical bonds are edges connecting the atoms, which are
1375 represented by nodes.",
1376 "Atoms form the nodes of the graph, while bonds form the edges.",
1377 "A representation of a molecule's atoms (nodes) and their bonds
1378 (edges).",
1379 "This graph depicts a molecule's atomic structure using nodes for
1380 atoms, edges for bonds.",
1381 "Vertices are atoms in a molecule, and edges are the bonds between
1382 them.",
1383 "A molecular graph where nodes stand for atoms and edges for
1384 chemical bonds.",
1385 "The graph's nodes are atoms; its edges are the bonds linking
1386 them.",
1387 "Modeling a chemical compound with atoms as nodes and bonds as
1388 connecting edges.",
1389 "A graph where nodes are atoms and edges show their covalent
1390 connections.",
1391 "Each node signifies an atom, and each edge signifies a chemical
1392 bond.",
1393 "This graph maps atoms to nodes and their chemical bonds to
1394 edges.",
1395 "A molecular representation using nodes for atoms and edges for
1396 their bonds.",
1397 "Atoms are modeled as nodes, and their covalent bonds are modeled
1398 as edges.",
1399 "A graph where vertices represent atoms and edges represent
1400 chemical bonds.",
1401 "This model uses nodes for atoms and edges for the bonds between
1402 them.",
1403 "Depicts a molecule where atoms are vertices and bonds are
1404 edges.",
1405 "A chemical graph with atoms as nodes and their connections as
1406 edges.",
1407 "Nodes are atoms, and the edges show the bonds that connect
1408 them.",

1404 "This graph's vertices are atoms, and its edges are chemical
1405 bonds.",
1406 "A representation where atoms are nodes and the edges are their
1407 covalent bonds.",
1408 "Molecular structure where atoms are nodes and edges are their
1409 connecting bonds.",
1410 "Graph showing atoms as nodes and the chemical bonds between them
1411 as edges.",
1412 "Atoms are represented by nodes, and bonds are represented by
1413 edges.",
1414 "A graph of a molecule where nodes are atoms and edges are
1415 bonds.",
1416 "The vertices of this graph are atoms; the edges are their
1417 chemical bonds.",
1418 "This graph illustrates a molecule's atoms as nodes and its bonds
1419 as edges.",
1420 "A model where nodes denote atoms and edges denote covalent
1421 bonds.",
1422 "Represents a molecule's atoms and bonds as a graph's nodes and
1423 edges.",
1424 "Each node is an atom, connected to others by edges representing
1425 bonds.",
1426 "A graph structure where nodes are atoms and edges are chemical
1427 bonds.",
1428 "The molecule's atoms are nodes, and the bonds between them are
1429 edges.",
1430 "A graph where each vertex is an atom and each edge a covalent
1431 bond.",
1432 "This graph's nodes correspond to atoms, and its edges to chemical
1433 bonds.",
1434 "A molecular structure is shown, with nodes as atoms and edges as
1435 bonds.",
1436 "Atoms are the vertices, and the bonds between them are the
1437 edges.",
1438 "A graph in which nodes are atoms and edges are their chemical
1439 bonds.",
1440 "This represents a molecule's structure with nodes for atoms,
1441 edges for bonds.",
1442 "Nodes stand for atoms, while edges stand for the bonds between
1443 them.",
1444 "A graph depicting atoms as nodes and their covalent bonds as
1445 edges.",
1446 "The atoms of a molecule are nodes; their bonds are the graph's
1447 edges.",
1448 "A model of a molecule using vertices for atoms and edges for
1449 bonds.",
1450 "This graph shows atoms as nodes and their bonds as connecting
1451 edges.",
1452 "A chemical structure where atoms are vertices and bonds are
1453 edges.",
1454 "Nodes are the atoms, and edges are the chemical bonds they
1455 form.",
1456 "Represents a molecule with atoms as vertices and bonds as
1457 edges.",
1458 "A graph where nodes are atoms and edges are the links between
1459 them.",
1460 "The vertices in this graph are atoms, and the edges are covalent
1461 bonds.",

1458 "A molecular graph where atoms are nodes and their bonds are the
1459 edges.",
1460 "This graph models atoms as vertices and their chemical bonds as
1461 edges.",
1462 "A representation where nodes are atoms and edges are their
1463 chemical connections.",
1464 "Atoms are depicted as nodes, and their bonds are depicted as
1465 edges.",
1466 "A graph illustrating a molecule with atoms as nodes and bonds as
1467 edges.",
1468 "Nodes represent the atoms, and edges represent the bonds in a
1469 molecule.",
1470 "This graph's nodes are a molecule's atoms, and its edges are the
1471 bonds.",
1472 "A model where vertices are atoms and edges are their covalent
1473 bonds.",
1474 "Depicts a chemical structure with nodes for atoms and edges for
1475 bonds.",
1476 "Atoms are the nodes, and the edges are the bonds that link
1477 them.",
1478 "A graph where each node is an atom and each edge is a bond.",
1479 "The vertices represent atoms, and the edges represent the bonds
1480 between them.",
1481 "A molecular graph with nodes as atoms and edges as chemical
1482 bonds.",
1483 "This graph shows a molecule's atoms as nodes and their bonds as
1484 edges.",
1485 "Nodes are atoms in a chemical structure; edges are the bonds.",
1486 "A graph where atoms are vertices and their covalent bonds are
1487 edges.",
1488 "This model shows atoms as nodes and the bonds connecting them as
1489 edges.",
1490 "Represents a molecule's atomic layout with nodes for atoms, edges
1491 for bonds.",
1492 "Each vertex corresponds to an atom, and each edge to a chemical
1493 bond.",
1494 "A graph of a molecule's atoms (vertices) and their bonds
1495 (edges).",
1496 "The nodes of this graph represent atoms, the edges represent
1497 their bonds.",
1498 "A molecular structure modeled with atoms as nodes and bonds as
1499 edges.",
1500 "This graph maps a molecule's atoms to nodes and its bonds to
1501 edges.",
1502 "A representation where vertices are atoms and edges are their
1503 covalent bonds.",
1504 "Atoms are shown as nodes, and their chemical bonds are shown as
1505 edges.",
1506 "A graph where the nodes are atoms and the edges are chemical
1507 bonds.",
1508 "This graph's vertices stand for atoms, and its edges stand for
1509 bonds.",
1510 "A model of a chemical structure with nodes as atoms, edges as
1511 bonds.",
1512 "Depicts atoms as nodes and the bonds that connect them as
1513 edges.",
1514 "A graph where each atom is a node and each bond is an edge.",
1515 "The atoms are the graph's nodes, and their bonds are the edges.",

1512 "This represents a molecule where atoms are nodes and bonds are
1513 edges.",
1514 "Nodes are atoms, and the edges between them represent covalent
1515 bonds.",
1516 "A graph showing a molecule's atoms as vertices and its bonds as
1517 edges.",
1518 "The vertices are atoms, and the edges are the chemical bonds
1519 connecting them.",
1520 "A molecular graph where nodes represent atoms and edges represent
1521 their bonds.",
1522 "This graph models a molecule's atoms as nodes and their bonds as
1523 edges.",
1524 "A representation of a chemical structure with atoms as nodes,
1525 bonds as edges.",
1526 "Atoms are mapped to nodes, and their bonds are mapped to edges.",
1527 "A graph where nodes are atoms and edges are the connections
1528 between them.",
1529 "The vertices of the graph are atoms, and the edges are their
1530 bonds.",
1531 "Edges represent bonds between atoms, which are represented by
1532 nodes.",
1533 "Bonds are shown as edges, connecting the atoms which are the
1534 nodes.",
1535 "This graph's edges are bonds, and its nodes are the atoms they
1536 connect.",
1537 "Covalent bonds are edges, linking the atoms which are the
1538 vertices.",
1539 "Each edge is a chemical bond, and each vertex is an atom.",
1540 "A graph where edges are bonds and nodes are the atoms they
1541 link.",
1542 "The bonds are edges, and the atoms they connect are the nodes.",
1543 "Edges are covalent bonds connecting the atoms, which are the
1544 nodes.",
1545 "This graph models bonds as edges and the atoms they link as
1546 nodes.",
1547 "Represents chemical bonds as edges and atoms as the connected
1548 nodes.",
1549 "The edges are bonds, and the vertices are the atoms in the
1550 molecule.",
1551 "A model where edges are bonds and nodes are the atoms they
1552 connect.",
1553 "Bonds are the edges, connecting the atoms which are the graph's
1554 nodes.",
1555 "Edges are chemical bonds, and the nodes they connect are atoms.",
1556 "This graph's edges represent bonds, and its nodes represent
1557 atoms.",
1558 "A representation where edges are bonds and nodes are the
1559 connected atoms.",
1560 "The chemical bonds are edges, and the atoms are the graph's
1561 vertices.",
1562 "Edges denote covalent bonds, and nodes denote the atoms they
1563 link.",
1564 "This graph shows bonds as edges and the atoms they connect as
1565 nodes.",
1566 "A structure where edges are bonds and the nodes are atoms.",
1567 "Bonds are modeled as edges, and the atoms they connect are
1568 nodes.",
1569 "The edges of this graph are bonds; the nodes are the atoms.",

1566 "A graph where bonds are edges and the atoms they connect are
1567 nodes.",
1568 "This model uses edges for bonds and nodes for the connected
1569 atoms.",
1570 "Depicts bonds as edges and the atoms they connect as vertices.",
1571 "A chemical graph with bonds as edges and atoms as the nodes.",
1572 "Edges are the bonds, and the nodes are the atoms they connect.",
1573 "This graph's edges are chemical bonds, and its vertices are
1574 atoms.",
1575 "A representation where bonds are edges and the connected atoms
1576 are nodes.",
1577 "Molecular bonds are edges, and the atoms they connect are
1578 nodes.",
1579 "Graph showing bonds as edges and the atoms they connect as
1580 nodes.",
1581 "Bonds are represented by edges, and atoms are represented by
1582 nodes.",
1583 "A graph of a molecule where edges are bonds and nodes are
1584 atoms.",
1585 "The edges of this graph are chemical bonds; the vertices are
1586 atoms.",
1587 "This graph illustrates bonds as edges and its atoms as nodes.",
1588 "A model where edges denote bonds and nodes denote atoms.",
1589 "Represents a molecule's bonds and atoms as a graph's edges and
1590 nodes.",
1591 "Each edge is a bond, connecting atoms which are represented by
1592 nodes.",
1593 "A graph structure where edges are bonds and nodes are atoms.",
1594 "The molecule's bonds are edges, and the atoms they connect are
1595 nodes.",
1596 "A graph where each edge is a covalent bond and each vertex is an
1597 atom.",
1598 "This graph's edges correspond to bonds, and its nodes to atoms.",
1599 "A molecular structure is shown, with edges as bonds and nodes as
1600 atoms.",
1601 "Bonds are the edges, and the atoms they connect are the
1602 vertices.",
1603 "A graph in which edges are chemical bonds and nodes are atoms.",
1604 "This represents a molecule's structure with edges for bonds,
1605 nodes for atoms.",
1606 "Edges stand for bonds, while nodes stand for the atoms they
1607 connect.",
1608 "A graph depicting bonds as edges and their connected atoms as
1609 nodes.",
1610 "The bonds of a molecule are edges; their atoms are the graph's
1611 nodes.",
1612 "A model of a molecule using edges for bonds and vertices for
1613 atoms.",
1614 "This graph shows bonds as edges and their connected atoms as
1615 nodes.",
1616 "A chemical structure where bonds are edges and atoms are
1617 vertices.",
1618 "Edges are the chemical bonds, and nodes are the atoms they
1619 form.",
1620 "Represents a molecule with bonds as edges and atoms as
1621 vertices.",
1622 "A graph where edges are the links and nodes are the atoms.",
1623 "The edges in this graph are covalent bonds, and the vertices are
1624 atoms.",

1620 "A molecular graph where bonds are edges and their atoms are the
1621 nodes.",
1622 "This graph models bonds as edges and their chemical atoms as
1623 vertices.",
1624 "A representation where edges are chemical connections and nodes
1625 are atoms.",
1626 "Bonds are depicted as edges, and their atoms are depicted as
1627 nodes.",
1628 "A graph illustrating a molecule with bonds as edges and atoms as
1629 nodes.",
1630 "Edges represent the bonds, and nodes represent the atoms in a
1631 molecule.",
1632 "This graph's edges are a molecule's bonds, and its nodes are the
1633 atoms.",
1634 "A model where edges are covalent bonds and vertices are atoms.",
1635 "Depicts a chemical structure with edges for bonds and nodes for
1636 atoms.",
1637 "Bonds are the edges, and the nodes are the atoms that they
1638 link.",
1639 "A graph where each edge is a bond and each node is an atom.",
1640 "The edges represent the bonds, and the vertices represent the
1641 atoms.",
1642 "A molecular graph with edges as chemical bonds and nodes as
1643 atoms.",
1644 "This graph shows a molecule's bonds as edges and their atoms as
1645 nodes.",
1646 "Edges are the bonds in a chemical structure; nodes are the
1647 atoms.",
1648 "A graph where bonds are edges and their covalent atoms are
1649 vertices.",
1650 "This model shows bonds as edges and the atoms they connect as
1651 nodes.",
1652 "Represents a molecule's bond layout with edges for bonds, nodes
1653 for atoms.",
1654 "Each edge corresponds to a chemical bond, and each vertex to an
1655 atom.",
1656 "A graph of a molecule's bonds (edges) and their atoms
1657 (vertices).",
1658 "The edges of this graph represent bonds, the nodes represent
1659 their atoms.",
1660 "A molecular structure modeled with edges as bonds and nodes as
1661 atoms.",
1662 "This graph maps a molecule's bonds to edges and its atoms to
1663 nodes.",
1664 "A representation where edges are covalent bonds and vertices are
1665 atoms.",
1666 "Bonds are shown as edges, and their chemical atoms are shown as
1667 nodes.",
1668 "A graph where the edges are chemical bonds and the nodes are
1669 atoms.",
1670 "This graph's edges stand for bonds, and its vertices stand for
1671 atoms.",
1672 "A model of a chemical structure with edges as bonds, nodes as
1673 atoms.",
1674 "Depicts bonds as edges and the atoms that they connect as
1675 nodes.",
1676 "A graph where each bond is an edge and each atom is a node.",
1677 "The bonds are the graph's edges, and their atoms are the nodes.",

```
1674 "This represents a molecule where bonds are edges and atoms are
1675 nodes.",
1676 "Edges are covalent bonds, and the nodes between them represent
1677 atoms.",
1678 "A graph showing a molecule's bonds as edges and its atoms as
1679 vertices.",
1680 "The edges are the chemical bonds, and the vertices are the
1681 connecting atoms.",
1682 "A molecular graph where edges represent bonds and nodes represent
1683 their atoms.",
1684 "This graph models a molecule's bonds as edges and their atoms as
1685 nodes.",
1686 "A representation of a chemical structure with edges as bonds,
1687 nodes as atoms.",
1688 "Bonds are mapped to edges, and their atoms are mapped to nodes.",
1689 "A graph where edges are the connections and nodes are the
1690 atoms.",
1691 "The edges of the graph are bonds, and the vertices are their
1692 atoms."
1693 ]
```

1694 H.2 IMDB-* GRAPHS DESCRIPTIONS DATASET

```
1695 [
1696
1697 "Nodes are actors; an edge connects them if they appeared in the
1698 same movie together.",
1699 "An edge links two actor nodes if they both shared the screen in
1700 the same film.",
1701 "Actors who have co-starred in a movie are connected by an edge
1702 in this graph.",
1703 "This graph connects actors as nodes, an edge signifying a
1704 shared movie credit between them.",
1705 "When two actors appear in the same film, their corresponding
1706 nodes are linked by an edge.",
1707 "Vertices represent actors; a connection is formed if they
1708 collaborated on the same film project.",
1709 "Nodes are actors. An edge exists if two actors have worked on a
1710 movie together.",
1711 "An edge connects two nodes if the actors they represent
1712 appeared in the same motion picture.",
1713 "This network models actors as nodes, with edges for sharing a
1714 credit in a movie.",
1715 "If two actors were cast in the same movie, their nodes are
1716 connected by an edge.",
1717 "Actors are nodes. A link is drawn between them if they shared a
1718 film credit.",
1719 "Connect two actor nodes with an edge if they have ever
1720 performed in a movie together.",
1721 "This graph represents actors as nodes, linking those who have
1722 co-appeared in any film.",
1723 "A shared movie appearance creates an edge between the nodes
1724 representing the respective actors.",
1725 "Nodes are performers, and an edge links them if they acted in
1726 the same film production.",
1727 "An edge is created between two actor nodes if they were both in
1728 the same movie.",
1729 "The vertices are actors; an edge connects them if they were
1730 co-stars in a film.",
```

1728 "Actors are represented as nodes, connected by edges if they
1729 appeared in a shared movie.",
1730 "Draw an edge between two actor nodes if they have a common film
1731 in their filmography.",
1732 "A connection exists between two actor nodes for a shared movie
1733 appearance.",
1734 "Nodes symbolize actors. An edge signifies that they have acted
1735 together in a movie.",
1736 "If actors share a movie credit, their nodes are connected in
1737 this graph representation.",
1738 "This is a graph of actors, where an edge connects any two who
1739 acted together.",
1740 "Two actor nodes are linked if the performers appeared together
1741 in the same feature film.",
1742 "An edge is present between two nodes if the actors shared a
1743 role in a movie.",
1744 "Actors who have collaborated on a film are connected by an edge
1745 between their nodes.",
1746 "Nodes represent actors. A link is formed if they have ever
1747 co-starred in a film.",
1748 "This network connects actors who have appeared in at least one
1749 movie together.",
1750 "An edge between two nodes means the actors performed in the
1751 same film.",
1752 "Let nodes be actors. An edge connects them if they shared a
1753 movie production.",
1754 "A graph where nodes are actors and an edge means a shared movie
1755 credit.",
1756 "Actors are nodes, linked by an edge if they appeared in the
1757 same motion picture.",
1758 "An edge joins two actor nodes if they have ever been in a movie
1759 together.",
1760 "Vertices are actors. A link between them means they acted in
1761 the same movie.",
1762 "Connect nodes if the actors they represent were featured in the
1763 same movie.",
1764 "Nodes are actors. They are connected if they have ever shared a
1765 movie set.",
1766 "An edge exists between actor nodes for every shared film
1767 appearance.",
1768 "The graph links actors as nodes if they have a common movie
1769 credit.",
1770 "Actors are the nodes, and a shared film role creates an edge
1771 between them.",
1772 "A shared movie credit between two actors results in an edge
1773 connecting their nodes.",
1774 "Nodes are actors. An edge connects two if they have a shared
1775 movie in common.",
1776 "If two actors were in the same movie, their nodes are connected
1777 by a link.",
1778 "This graph's nodes are actors. Edges connect those who have
1779 acted in a movie together.",
1780 "An edge connects two actor nodes if they have ever shared a
1781 screen credit.",
1782 "Actors are nodes. A connection is made if they have appeared in
1783 a film together.",
1784 "The nodes are actors, and an edge means they worked on the same
1785 movie.",

1782 "An edge links actor nodes that have co-appeared in any film
1783 production.",
1784 "Nodes represent actors. An edge is drawn for a shared
1785 appearance in a movie.",
1786 "Two nodes are connected if the actors they represent have
1787 co-starred in a motion picture.",
1788 "This network connects actors as nodes if they have ever worked
1789 on a movie together.",
1790 "A link is established between actor nodes if they were cast in
1791 the same film.",
1792 "The vertices are actors, connected by an edge if they were in
1793 the same movie.",
1794 "An edge exists between two nodes if their actors appeared in
1795 the same film.",
1796 "Nodes are actors. An edge means they have a common movie credit
1797 on their resumes.",
1798 "Connect two actor nodes if they have at any point acted in a
1799 movie together.",
1800 "A shared appearance in a film creates an edge between the
1801 corresponding actor nodes.",
1802 "This graph models actors as nodes, connecting them if they
1803 co-starred in a movie.",
1804 "An edge links actors who have worked together on any single
1805 movie production.",
1806 "Nodes represent actors. An edge signifies they were featured in
1807 the same motion picture.",
1808 "If actors have a shared movie credit, we connect their nodes
1809 with an edge.",
1810 "This is a network of actors, linked together if they appeared
1811 in a movie.",
1812 "Vertices are actors. An edge between them indicates a shared
1813 film appearance.",
1814 "Nodes are actors. A shared movie appearance between them
1815 results in a connecting edge.",
1816 "Connect two actor nodes if the performers have ever shared the
1817 screen in a film.",
1818 "An edge is created between actor nodes if they have a mutual
1819 movie credit.",
1820 "This graph links actors who have been cast in the same movie at
1821 any time.",
1822 "Nodes are actors. An edge connects them if they have a shared
1823 film in their history.",
1824 "A connection is made between two actor nodes if they acted in
1825 the same movie.",
1826 "If two actors worked together on a movie, their nodes are
1827 linked in this graph.",
1828 "An edge connects nodes of actors who have been co-stars in any
1829 given movie.",
1830 "The nodes are actors, and they are linked if they appeared in a
1831 movie together.",
1832 "Actors who have shared a film set are connected by an edge
1833 between their nodes.",
1834 "This network's nodes are actors. An edge means they appeared in
1835 the same movie.",
1836 "An edge links two actor nodes when they have a common movie
1837 credit.",
1838 "Nodes represent actors. An edge is formed for a shared role in
1839 a film.",
1840

1836 "Connect nodes representing actors who have collaborated on a
1837 motion picture.",
1838 "An edge exists between two nodes if the actors have ever acted
1839 together.",
1840 "The graph has actors as nodes, with edges for shared movie
1841 appearances.",
1842 "A link between two nodes means the actors have a shared movie
1843 credit.",
1844 "Nodes are actors. An edge indicates they have been in a film
1845 together.",
1846 "Two actor nodes are connected if they have a mutual film in
1847 their careers.",
1848 "This graph connects actors as nodes if they have ever been in a
1849 movie together.",
1850 "An edge signifies that two actors have co-starred in the same
1851 film production.",
1852 "Vertices are actors. A connection exists if they were featured
1853 in the same movie.",
1854 "Nodes are actors. An edge is drawn if they have acted in a
1855 movie together.",
1856 "If two actors have a shared film credit, their nodes are
1857 connected with an edge.",
1858 "The network's nodes are actors, linked if they were cast in the
1859 same movie.",
1860 "An edge connects two actor nodes if they have ever worked on a
1861 film together.",
1862 "A shared movie role between two actors creates an edge
1863 connecting their nodes.",
1864 "This graph represents actors as nodes, linked if they have
1865 shared a movie credit.",
1866 "Nodes are actors. An edge is present if they have ever been in
1867 a movie together.",
1868 "Connect two nodes if the actors they represent have ever been
1869 in a film together.",
1870 "An edge links nodes of actors who have co-appeared in the same
1871 movie.",
1872 "A connection between actor nodes is formed if they have a
1873 shared movie credit.",
1874 "Nodes are actors. A link is created if they acted in the same
1875 movie.",
1876 "If two actors have worked on a film together, their nodes are
1877 connected by an edge.",
1878 "The vertices represent actors, connected by edges if they
1879 shared a movie credit.",
1880 "This graph connects actors as nodes who have ever been in a
1881 movie together.",
1882 "An edge between two nodes indicates a shared movie credit for
1883 the actors.",
1884 "Nodes are actors. They are connected if they have ever acted in
1885 a film together.",
1886 "An edge exists between two actor nodes if they have a shared
1887 film in their filmography.",
1888 "Connect nodes if their actors have appeared together in any
1889 movie.",
1890 "A link between two actor nodes means they have a mutual movie
1891 credit.",
1892 "This network has actors as nodes, with edges for co-starring in
1893 a film.",

1890 "Nodes are actors. An edge signifies that they have been in a
1891 movie together.",
1892 "Two nodes are connected if their respective actors have ever
1893 been in a movie together.",
1894 "An edge links actor nodes if they have ever collaborated on a
1895 film project.",
1896 "The graph's nodes are actors, connected if they have shared a
1897 movie credit.",
1898 "A shared appearance in a movie results in an edge between the
1899 actor nodes.",
1900 "Nodes are actors. An edge connects them if they have a mutual
1901 movie appearance.",
1902 "If two actors have been in a movie together, their nodes are
1903 connected by an edge.",
1904 "This graph links actors who have ever worked on the same movie
1905 production.",
1906 "An edge connects two actor nodes if they share a credit in any
1907 film.",
1908 "Vertices are actors. A link is made if they have been in a
1909 movie together.",
1910 "Nodes represent actors. An edge is created if they have a
1911 shared movie credit.",
1912 "Connect two nodes if the actors have ever been cast in the same
1913 movie.",
1914 "An edge means that two actors have at some point been in a
1915 movie together.",
1916 "The network connects actors as nodes if they have ever shared a
1917 film credit.",
1918 "A connection is established between actor nodes for a shared
1919 movie role.",
1920 "Nodes are actors. An edge is formed if they have ever been in a
1921 film together.",
1922 "If two actors have appeared in a movie together, their nodes
1923 are linked.",
1924 "This graph's vertices are actors, connected by edges for shared
1925 movie roles.",
1926 "An edge links two nodes if the actors have ever been in a movie
1927 together.",
1928 "Nodes are actors. A connection is made if they have a shared
1929 movie credit.",
1930 "Two nodes are linked if the actors have ever been in the same
1931 movie.",
1932 "An edge signifies that two actors have a shared credit in their
1933 filmographies.",
1934 "The graph connects actors who have at any time been in a movie
1935 together.",
1936 "A link between two nodes means the actors have been in a movie
1937 together.",
1938 "Nodes are actors. An edge is drawn between them if they have a
1939 shared movie.",
1940 "If actors have been in a movie together, their nodes are
1941 connected by an edge.",
1942 "This network represents actors as nodes, connected if they have
1943 a shared film credit.",
1944 "An edge between two nodes means the actors have a shared movie
1945 credit.",
1946 "Vertices are actors. A connection is made if they have acted in
1947 a movie together.",

1944 "Nodes are actors. An edge connects them if they have ever
1945 worked together on a film.",
1946 "Connect two actor nodes if they have ever been in the same film
1947 production.",
1948 "An edge links nodes if the actors have a shared movie
1949 appearance.",
1950 "The graph has actors as nodes. An edge means they were in a
1951 movie together.",
1952 "A shared film credit creates an edge between the corresponding
1953 actor nodes.",
1954 "Nodes are actors. A link is made if they have ever been in a
1955 movie together.",
1956 "If two actors have a shared film, an edge connects their
1957 nodes.",
1958 "This network connects actors as nodes if they have ever been in
1959 a movie together.",
1960 "An edge signifies that two actors have been in the same movie
1961 together.",
1962 "Vertices are actors. A connection is made if they have a shared
1963 movie credit.",
1964 "Nodes are actors. An edge is created if they have ever been in
1965 a movie together.",
1966 "Connect two nodes if their actors have a shared movie credit.",
1967 "An edge means that two actors have been in a movie together.",
1968 "The network connects actors who have ever been in a movie
1969 together as nodes.",
1970 "A connection is established between nodes if the actors were in
1971 the same movie.",
1972 "Nodes are actors. An edge is formed if they have been in a
1973 movie together.",
1974 "If two actors have been in a movie together, an edge connects
1975 their nodes.",
1976 "This graph's vertices are actors. Edges connect those with a
1977 shared movie credit.",
1978 "An edge links two nodes if their actors have ever been in a
1979 movie together.",
1980 "Nodes are actors. A connection is made if they share a movie
1981 credit.",
1982 "Two nodes are linked if their actors have ever been in the same
1983 movie.",
1984 "An edge signifies that two actors share a credit in their
1985 filmographies.",
1986 "The graph connects actors who have ever been in a movie
1987 together.",
1988 "A link between two nodes means the actors have been in a movie
1989 together.",
1990 "Nodes are actors. An edge is drawn if they have a shared
1991 movie.",
1992 "If actors have been in a movie together, their nodes are
1993 connected.",
1994 "This network represents actors as nodes, connected by a shared
1995 film credit.",
1996 "An edge between two nodes means the actors have a shared movie
1997 credit.",
1998 "Vertices are actors. A connection means they acted in a movie
1999 together.",
2000 "Nodes are actors. An edge connects them if they worked together
2001 on a film.",

1998 "Connect two actor nodes if they were in the same film
1999 production.",
2000 "An edge links nodes if the actors have a shared movie
2001 appearance.",
2002 "The graph has actors as nodes, with edges for being in a movie
2003 together.",
2004 "A shared film credit creates an edge between actor nodes.",
2005 "Nodes are actors. A link is made if they were in a movie
2006 together.",
2007 "If two actors have a shared film, an edge connects their
2008 nodes.",
2009 "This network connects actors as nodes if they were in a movie
2010 together.",
2011 "An edge signifies that two actors were in the same movie
2012 together.",
2013 "Vertices are actors. A connection is made if they have a shared
2014 movie credit.",
2015 "Nodes are actors. An edge is created if they were in a movie
2016 together.",
2017 "Connect two nodes if their actors have a shared movie credit.",
2018 "An edge means that two actors were in a movie together.",
2019 "The network connects actors who were in a movie together as
2020 nodes.",
2021 "A connection is established between nodes if the actors were in
2022 the same movie.",
2023 "Nodes are actors. An edge is formed if they were in a movie
2024 together.",
2025 "If two actors were in a movie together, an edge connects their
2026 nodes.",
2027 "This graph's vertices are actors. Edges connect those with a
2028 shared movie credit.",
2029 "An edge links two nodes if their actors were in a movie
2030 together.",
2031 "Nodes are actors. A connection is made if they share a movie
2032 credit.",
2033 "Two nodes are linked if their actors were in the same movie.",
2034 "An edge signifies that two actors share a credit in their
2035 filmographies.",
2036 "The graph connects actors who were in a movie together.",
2037 "A link between two nodes means the actors were in a movie
2038 together.",
2039 "Nodes are actors. An edge is drawn if they have a shared
2040 movie.",
2041 "If actors were in a movie together, their nodes are
2042 connected.",
2043 "This network represents actors as nodes, connected by a shared
2044 film credit.",
2045 "An edge between two nodes means the actors have a shared movie
2046 credit.",
2047 "Vertices are actors. A connection means they acted in a movie
2048 together.",
2049 "Nodes are actors. An edge connects them if they worked together
2050 on a film.",
2051 "Connect two actor nodes if they were in the same film
production.",
"An edge links nodes if the actors have a shared movie
appearance.",
"The graph has actors as nodes, with edges for being in a movie
together.",

2052 "A shared film credit creates an edge between actor nodes.",
2053 "Nodes are actors. A link is made if they were in a movie
2054 together."
2055]
2056
2057
2058 **H.3 REDDIT-* GRAPHS DESCRIPTIONS DATASET**
2059
2060 [
2061 "This network maps comment-based relationships between users.
2062 Each node is a user, and a link signifies a direct reply.",
2063 "Construct a graph where participants are nodes. A tie exists if
2064 one member commented on another.",
2065 "An edge connects two user nodes when a comment is exchanged
2066 between the individuals they represent.",
2067 "We model a social graph where vertices are users. A directed
2068 arc from user A to B indicates A commented on B's content.",
2069 "Individuals are represented as nodes. A connection is formed
2070 between two nodes through a comment.",
2071 "This graph illustrates user engagement. Nodes represent users,
2072 and edges show who is commenting on whom.",
2073 "Each node corresponds to a platform user. Edges indicate a
2074 direct comment was made from one to another.",
2075 "A user is a node. We draw an edge between two nodes if a
2076 comment interaction occurs between them.",
2077 "This graph connects authors who comment on each other's work,
2078 where each author is a node in the network.",
2079 "Nodes are users. An edge is created if one user writes a remark
2080 to another.",
2081 "An edge is formed between two user nodes if a comment is
2082 exchanged. The nodes themselves simply represent the users.",
2083 "Let nodes be contributors. An edge connects nodes if a comment
2084 is made between them.",
2085 "Users are modeled as nodes. An edge between two nodes signifies
2086 a direct comment interaction.",
2087 "The network's vertices are users. An edge connects two vertices
2088 if one user commented on the other's content.",
2089 "Define nodes as users. An edge represents a comment from one
2090 user to another.",
2091 "A comment from one user to another creates an edge between
2092 their corresponding nodes in the graph.",
2093 "Nodes are individual users. An edge appears if one user has
2094 commented on the other.",
2095 "Represent users as nodes. An edge signifies that one of the
2096 users commented on the other.",
2097 "This is a graph of users where an edge connects two users if
2098 one has replied to the other.",
2099 "Vertices in the graph are users. A connection exists if one
2100 user commented directly on the other's post.",
2101 "When a user comments on another's content, an edge is created
2102 between their corresponding nodes.",
2103 "Nodes are users. An edge links two nodes if a comment occurred
2104 between those users.",
2105 "The graph maps user interactions. Nodes are users and edges are
formed when one user comments on another.",
"Each node is a user. An edge forms if one user comments on
content created by the other.",
"Model a network where users are nodes. A comment between two
users creates an edge.",

2106 "An edge is created between two user nodes upon a comment
2107 interaction.",
2108 "Let each user be a node. An edge exists if one user commented
2109 on the other.",
2110 "User nodes are connected by an edge if a comment was posted
2111 from one to the other.",
2112 "The graph represents users as nodes. An edge is present if a
2113 comment was made from one user to another.",
2114 "Nodes correspond to users, and edges represent comment
2115 interactions between them.",
2116 "A directed edge connects a commenter to a recipient, both of
2117 whom are nodes representing users.",
2118 "We construct a graph where nodes are users and an edge
2119 indicates a comment was made.",
2120 "Each node is a user. An edge connects two nodes if a comment
2121 was exchanged between them.",
2122 "Users are nodes in a graph. A link is formed if one user
2123 replies to the other via comment.",
2124 "This network connects users through comments. A node is a user
2125 and an edge is a comment between them.",
2126 "Nodes symbolize users. Edges are formed based on direct comment
2127 interactions between these users.",
2128 "An interaction via comment creates an edge between two user
2129 nodes.",
2130 "The vertices are users. An edge links two users if one has
2131 commented on the other.",
2132 "Users are represented as nodes in a graph, and an edge
2133 signifies a comment interaction.",
2134 "A graph is built where nodes are users. A comment from user A
2135 to B creates a directed edge.",
2136 "Nodes are users. An edge is created if one user directs a
2137 comment at the other.",
2138 "A comment between users forms an edge connecting their nodes.",
2139 "Represent each user as a node. Connect two nodes with an edge
2140 if one user commented on the other.",
2141 "In this graph, a node is a user. An edge connects two users if
2142 one has commented on the other.",
2143 "We define a user interaction graph where nodes are users and
2144 edges represent comments.",
2145 "The graph's nodes are users. An edge is drawn between nodes if
2146 their users had a comment interaction.",
2147 "An edge signifies a comment from one user to another, where
2148 each user is a node.",
2149 "Let nodes be users. An edge from node A to B indicates user A
2150 commented on user B.",
2151 "Users are represented as nodes, and a comment between them
2152 forms an edge.",
2153 "The network's nodes are users. An edge exists if one user has
2154 posted a comment to the other.",
2155 "Nodes represent users. An edge connects two nodes if one user
2156 has commented to the other.",
2157 "A comment from one user to another establishes a connection
2158 between their nodes.",
2159 "In this representation, users are nodes. An edge is formed if
one user commented on another's content.",
"The graph links users who have commented on each other. The
nodes in the graph are the users themselves.",
"Each user is a node. A directed edge indicates a comment from
one user to another.",

2160 "Nodes are users. An edge between two nodes means one commented
2161 on the other's post.",
2162 "An edge connects two users if a comment was made between them.
2163 Each user is a node.",
2164 "We create a graph where nodes are users, and edges are the
2165 comments made between them.",
2166 "Users are nodes. An edge is present if one user comments on
2167 content posted by the other user.",
2168 "This graph models comment interactions. Nodes are users, and an
2169 edge means a comment was made between them.",
2170 "A node represents a user. An edge is created when a comment is
2171 directed from one user to another.",
2172 "Nodes are users. An edge between two nodes indicates a comment
2173 was exchanged.",
2174 "Let users be nodes. A directed edge connects a commenter to the
2175 recipient of the comment.",
2176 "The graph's vertices are users, and an edge exists if a comment
2177 was made between them.",
2178 "An edge is drawn between two user nodes if one commented on the
2179 other.",
2180 "Nodes are users. An edge is added if one user comments on the
2181 other's activity.",
2182 "Users are nodes in this network. An edge signifies a comment
2183 interaction.",
2184 "We build a graph of users where an edge signifies a comment.",
2185 "Represent users as nodes. If one comments on another, an edge
2186 connects their nodes.",
2187 "This network connects users via comments, where nodes are users
2188 and edges are the comments.",
2189 "Each node is a user, and a comment between them creates an
2190 edge.",
2191 "Nodes are users, and edges denote comment interactions between
2192 them.",
2193 "A user is a node. A directed edge is created from a user who
2194 comments to the one who receives it.",
2195 "The graph structure has users as nodes, and a comment creates
2196 an edge between them.",
2197 "If user A comments on user B, an edge links their respective
2198 nodes.",
2199 "Nodes are users. An edge between them exists if one user
2200 commented on the other's post.",
2201 "The graph's nodes represent users. An edge connects two nodes
2202 if a comment was made between them.",
2203 "An edge indicates a comment from one user to another. The nodes
2204 in the graph are the users.",
2205 "Users are nodes. An edge is established when one user comments
2206 on another's content.",
2207 "This graph maps user interactions. Nodes are users and edges
2208 signify a comment between two users.",
2209 "A node is a user. An edge is formed if a comment is made by one
2210 user to another.",
2211 "Nodes represent users, and an edge is a comment between them.",
2212 "Users are modeled as nodes. An edge signifies a comment
2213 directed from one user to another.",
2214 "Vertices in the graph are users, and an edge connects them if a
2215 comment was exchanged.",
2216 "An edge connects two user nodes following a comment
2217 interaction.",

2214 "Nodes are users. A directed edge is drawn from the commenter to
2215 the one being commented on.",
2216 "We model a network where users are nodes, and comments are
2217 edges.",
2218 "A user is a node. An edge is drawn if one user has commented on
2219 the other.",
2220 "This graph connects users through their comments, where nodes
2221 are users.",
2222 "Nodes are users. An edge is created if one user directs a
2223 comment at another.",
2224 "An edge is formed between two nodes if their corresponding
2225 users had a comment interaction.",
2226 "Let nodes be users. An edge connects two nodes if one user
2227 posted a comment to the other.",
2228 "Users are the nodes of the graph, and a comment between them
2229 creates an edge.",
2230 "The network's nodes are users, and a comment forms an edge
2231 between them.",
2232 "A comment interaction between two users creates an edge between
2233 their nodes.",
2234 "Nodes are users. A link between two nodes means one user
2235 commented on the other.",
2236 "We construct a graph where users are nodes. A comment from one
2237 to another creates an edge.",
2238 "Each user is a node. An edge exists between nodes if their
2239 users exchanged a comment.",
2240 "Users are represented as nodes, and their comment interactions
2241 are represented as edges.",
2242 "The graph consists of user nodes. An edge is drawn if a comment
2243 is made between them.",
2244 "A directed edge from node A to B means user A commented on user
2245 B.",
2246 "Nodes represent users. An edge connects nodes if a comment was
2247 made between the users.",
2248 "An edge is created between two user nodes when one comments on
2249 the other.",
2250 "Let users be nodes. An edge is formed if one user comments on
2251 another's content.",
2252 "The graph models users as nodes and comments as edges.",
2253 "A user is a node, and an edge is a comment between two such
2254 nodes.",
2255 "Nodes are users. An edge is drawn when a comment is made from
2256 one user to another.",
2257 "This network's vertices are users, connected by edges if a
2258 comment was made.",
2259 "An edge links two nodes if one user commented on the other. The
2260 nodes represent users.",
2261 "Users are nodes in a graph. An edge is created if one user
2262 replies to another with a comment.",
2263 "The graph maps user comments. Nodes are users and edges are
2264 formed when one user comments on another.",
2265 "A node is a user. An edge is established if a comment
2266 interaction occurs between two users.",
2267 "Nodes represent users, and edges signify comment
interactions.",
"Users are modeled as nodes. A directed edge connects a
commenter to the comment's recipient.",
"A graph is built with users as nodes and comments as edges.",

2268 "A comment from user A to user B creates a link between their
2269 nodes.",
2270 "Nodes are users. An edge is present if one user has commented
2271 on another's activity.",
2272 "The graph's nodes are users. An edge is drawn if a comment
2273 interaction happened between them.",
2274 "An edge signifies a comment between two users, who are the
2275 nodes of the graph.",
2276 "Users are nodes. An edge appears when one user directs a
2277 comment at another user.",
2278 "This graph connects users who have engaged via comments. Nodes
2279 are users and edges are comments.",
2280 "A node is a user. An edge forms when one user comments on
2281 another's content.",
2282 "Nodes represent users, and an edge indicates a comment was made
2283 between them.",
2284 "Let users be nodes. A directed edge is created when one user
2285 comments on another's post.",
2286 "The vertices of the graph are users, and an edge signifies a
2287 comment.",
2288 "An edge connects two user nodes if a comment was exchanged
2289 between them.",
2290 "Nodes are users. An edge is added if one user comments on
2291 another user's post.",
2292 "This network shows user comment interactions. Nodes are users
2293 and an edge means one commented on another.",
2294 "We build a graph of users, where an edge represents a comment
2295 between them.",
2296 "Represent users as nodes. If one user comments on another,
2297 their nodes are connected.",
2298 "This network maps user comments. Nodes are users and an edge is
2299 formed for each comment between users.",
2300 "Each node is a user, and a comment creates an edge between
2301 them.",
2302 "Nodes are users, and edges represent comment-based
2303 interactions.",
2304 "A user is a node. A directed edge is created for a comment from
2305 one user to another.",
2306 "The graph structure uses users as nodes. A comment forms an
2307 edge between them.",
2308 "When user A comments on user B, an edge links their respective
2309 nodes.",
2310 "Nodes are users. An edge exists if one user commented on
2311 another's contribution.",
2312 "The graph's nodes are users. An edge is made between nodes if a
2313 comment was exchanged.",
2314 "An edge indicates a comment from one user to another. The nodes
2315 are users.",
2316 "Users are nodes. An edge is formed when one user comments on
2317 something another user posted.",
2318 "This graph connects users who comment on one another. Nodes are
2319 users, and edges signify a comment interaction.",
2320 "A node is a user. An edge is formed if a comment is directed
2321 from one user to another.",
2322 "Nodes represent users, and an edge is a comment.",
2323 "Users are modeled as nodes. An edge is created if one user
2324 comments on another's post.",
2325 "Vertices in the graph are users, and an edge connects them if a
2326 comment was made.",

2322 "An edge connects two user nodes after a comment is made between
2323 them.",
2324 "Nodes are users. A directed edge goes from the commenter to the
2325 person they commented on.",
2326 "We model a network where users are nodes, and directed comments
2327 are edges.",
2328 "A user is a node. An edge is drawn if one user has commented on
2329 another's content.",
2330 "This graph links users via their comments. Nodes are users and
2331 edges represent a comment between them.",
2332 "Nodes are users. An edge is created if one user posts a comment
2333 to another.",
2334 "An edge is formed between two nodes if their users had a
2335 comment exchange.",
2336 "Let nodes be users. An edge connects two nodes if one user
2337 wrote a comment to the other.",
2338 "Users are the nodes of this graph, and a comment interaction
2339 creates an edge.",
2340 "The network's nodes are users, and a comment between them makes
2341 an edge.",
2342 "A comment interaction between users creates a link between
2343 their nodes.",
2344 "Nodes are users. A connection between two nodes means one user
2345 commented to the other.",
2346 "We construct a graph where users are nodes and a comment from
2347 one to another creates a directed edge.",
2348 "Each user is a node. An edge exists between nodes if a comment
2349 was made between them.",
2350 "Users are represented as nodes, and their direct comments as
2351 edges.",
2352 "The graph consists of user nodes. An edge is drawn if a comment
2353 is posted between them.",
2354 "A directed edge from node A to B indicates that user A
2355 commented on user B, where A and B are users.",
2356 "Nodes represent users. An edge connects nodes if a comment was
2357 posted between the users.",
2358 "An edge is created between two user nodes when one comments on
2359 the other's post.",
2360 "Let users be nodes. An edge is formed if one user comments on
2361 another person's content.",
2362 "The graph models users as nodes and directed comments as
2363 edges.",
2364 "A user is a node, and an edge is a comment interaction between
2365 two user nodes.",
2366 "Nodes are users. An edge is drawn when a comment is made from
2367 one user to another.",
2368 "This network's vertices are users, connected by edges if a
2369 comment was made.",
2370 "An edge links two nodes if one user commented on the other's
2371 content, with nodes being users.",
2372 "Users are nodes in a graph. An edge is created if one user
2373 replies to another.",
2374 "The graph maps user comment activity. Nodes are users and edges
2375 are formed when one comments on another.",
2376 "A node is a user. An edge is established if a comment
2377 interaction occurs between two of them.",
2378 "Nodes represent users, and edges signify direct comment
2379 interactions between them.",
2380

2376 "Users are modeled as nodes. A directed edge connects a
2377 commenter to the recipient of their comment.",
2378 "A graph is built with users as nodes and their comments as the
2379 edges between them.",
2380 "A comment from user A to user B creates a link between their
2381 corresponding nodes.",
2382 "Nodes are users. An edge is present if one user has commented
2383 on another user's activity.",
2384 "The graph's nodes are users. An edge is drawn if a comment
2385 interaction took place between them.",
2386 "An edge signifies a comment between two distinct users, who are
2387 represented as nodes.",
2388 "Users are nodes. An edge appears when one user directs a
2389 comment towards another user.",
2390 "This graph connects users who have engaged with each other via
2391 comments. Nodes are users, edges are comments.",
2392 "A node is a user. An edge forms when one user comments on
2393 another user's content.",
2394 "Nodes represent users, and an edge indicates a comment was made
2395 between them.",
2396 "Let users be nodes. A directed edge is created when one user
2397 comments on another user's post.",
2398 "The vertices of the graph are users, and an edge signifies a
2399 comment between them.",
2400 "An edge connects two user nodes if a comment was exchanged
2401 between those two users.",
2402 "Nodes are users. An edge is added if one user comments on
2403 another user's post.",
2404 "This network displays user comment interactions, where nodes
2405 are users.",
2406 "We build a graph of users, where an edge represents a
2407 comment.",
2408 "Represent users as nodes. If one user comments on another,
2409 their nodes are connected by an edge.",
2410 "This network maps user comments. Nodes are users and an edge is
2411 formed for each comment between them.",
2412 "Each node is a user. A comment creates an edge between two of
2413 them.",
2414 "Nodes are users, and edges represent comment-based interactions
2415 between them.",
2416 "A user is a node. A directed edge is created for a comment from
2417 one user to another.",
2418 "The graph structure uses users as nodes. A comment forms an
2419 edge between them.",
2420 "When user A comments on user B, an edge links their respective
2421 nodes.",
2422 "Nodes are users. An edge exists if one user commented on
2423 another's contribution to the platform.",
2424 "The graph's nodes are users. An edge is made between nodes if a
2425 comment was exchanged between them.",
2426 "An edge indicates a comment from one user to another user. The
2427 nodes are simply the users.",
2428 "Users are nodes. An edge is formed when one user comments on
2429 something another user has posted.",
2430 "This graph connects users who comment on one another's posts,
2431 where nodes are users.",
2432 "A node is a user. An edge is formed if a comment is directed
2433 from one user to another user.",

2430 "Nodes represent users, and an edge represents a comment between
2431 two users.",
2432 "Users are modeled as nodes. An edge is created if one user
2433 comments on another user's post.",
2434 "Vertices in the graph are users, and an edge connects them if a
2435 comment was made.",
2436 "An edge connects two user nodes after a comment is made between
2437 them.",
2438 "Nodes are users. A directed edge goes from the commenter to the
2439 person they commented on.",
2440 "We model a network where users are nodes, and directed comments
2441 form the edges.",
2442 "A user is a node. An edge is drawn if one user has commented on
2443 another."
2444]

2445 H.4 CITATION NETWORKS GRAPHS DESCRIPTIONS DATASET

2446
2447 [
2448 "The nodes correspond to academic papers. Edges reflect
2449 referenced literature connections.",
2450 "Nodes capture metadata about research manuscripts. Edges
2451 point to citation-based links.",
2452 "Each item represents a scholarly work. Edges tie together
2453 documents through citations.",
2454 "Edges indicate one paper citing another. These elements
2455 reflect authored academic contributions.",
2456 "These elements reflect authored academic contributions. Edges
2457 reflect referenced literature connections.",
2458 "Nodes denote individual research articles. Connections show
2459 references between documents.",
2460 "Each entry represents a scientific document. Edges reflect
2461 referenced literature connections.",
2462 "Nodes symbolize separate entries in a publication index.
2463 Links capture citation relationships.",
2464 "Nodes capture metadata about research manuscripts.
2465 Connections show references between documents.",
2466 "Each item represents a scholarly work. Connections show
2467 references between documents.",
2468 "The nodes mark distinct pieces of scientific writing. Links
2469 encode the act of citing prior work.",
2470 "Every node is tied to a research publication. Connections
2471 show references between documents.",
2472 "Nodes correspond to academic papers. Connections show
2473 references between documents.",
2474 "Nodes mark distinct pieces of scientific writing. Links
2475 encode the act of citing prior work.",
2476 "Each item represents a scholarly work. Connections show
2477 references between documents.",
2478 "Nodes denote individual research articles. Edges indicate one
2479 paper citing another.",
2480 "Every node is tied to a research publication. Links capture
2481 citation relationships.",
2482 "These nodes denote individual research articles. Links encode
2483 the act of citing prior work.",
2484 "Each item represents a scholarly work. Edges point to
2485 citation-based links.",
2486 "Nodes correspond to academic papers. Connections show
2487 references between documents.",

2484 "Nodes symbolize separate entries in a publication index.
2485 Edges reflect referenced literature connections.",
2486 "Nodes denote individual research articles. Links capture
2487 citation relationships.",
2488 "Each node holds information about a published study. Links
2489 encode the act of citing prior work.",
2490 "Nodes mark distinct pieces of scientific writing. Edges tie
2491 together documents through citations.",
2492 "Each entry represents a scientific document. Links capture
2493 citation relationships.",
2494 "Every node is tied to a research publication. Edges tie
2495 together documents through citations.",
2496 "Each item represents a scholarly work. Links encode the act
2497 of citing prior work.",
2498 "Nodes denote individual research articles. Links encode the
2499 act of citing prior work.",
2499 "Each entry represents a scientific document. Edges denote
2500 academic referencing paths.",
2501 "Each node holds information about a published study. Edges
2502 point to citation-based links.",
2503 "These elements reflect authored academic contributions. Links
2504 capture citation relationships.",
2505 "Each entry represents a scientific document. Edges reflect
2506 referenced literature connections.",
2507 "Nodes mark distinct pieces of scientific writing. Edges
2508 reflect referenced literature connections.",
2509 "Every node is tied to a research publication. Links encode
2510 the act of citing prior work.",
2510 "Nodes capture metadata about research manuscripts.
2511 Connections show references between documents.",
2512 "These elements reflect authored academic contributions.
2513 Connections arise from inter-paper citations.",
2514 "Nodes correspond to academic papers. Edges reflect referenced
2515 literature connections.",
2516 "Each node holds information about a published study. Edges
2517 indicate one paper citing another.",
2518 "Every node is tied to a research publication. Edges denote
2519 academic referencing paths.",
2520 "Nodes denote individual research articles. Connections arise
2521 from inter-paper citations.",
2522 "Each node holds information about a published study. Links
2523 capture citation relationships.",
2524 "Nodes symbolize separate entries in a publication index.
2525 Links capture citation relationships.",
2526 "Nodes correspond to academic papers. Edges denote academic
2527 referencing paths.",
2528 "Nodes denote individual research articles. Links encode the
2529 act of citing prior work.",
2529 "Each item represents a scholarly work. Connections arise from
2530 inter-paper citations.",
2531 "Every node is tied to a research publication. Links capture
2532 citation relationships.",
2533 "Nodes symbolize separate entries in a publication index.
2534 Connections show references between documents.",
2535 "Nodes denote individual research articles. Edges point to
2536 citation-based links.",
2537 "Each item represents a scholarly work. Connections show
2538 references between documents.",

2538 "Nodes mark distinct pieces of scientific writing. Edges
2539 reflect referenced literature connections.",
2540 "Nodes correspond to academic papers. Links encode the act of
2541 citing prior work.",
2542 "Each entry represents a scientific document. Connections show
2543 references between documents.",
2544 "Each node holds information about a published study. Links
2545 encode the act of citing prior work.",
2546 "Nodes denote individual research articles. Links encode the
2547 act of citing prior work.",
2548 "Nodes symbolize separate entries in a publication index.
2549 Edges denote academic referencing paths.",
2550 "Nodes capture metadata about research manuscripts. Links
2551 encode the act of citing prior work.",
2552 "Each item represents a scholarly work. Edges tie together
2553 documents through citations.",
2554 "Each node holds information about a published study. Edges
2555 denote academic referencing paths.",
2556 "Nodes correspond to academic papers. Edges point to
2557 citation-based links.",
2558 "These elements reflect authored academic contributions. Edges
2559 point to citation-based links.",
2560 "Each node holds information about a published study. Edges
2561 reflect referenced literature connections.",
2562 "Each entry represents a scientific document. Edges indicate
2563 one paper citing another.",
2564 "Nodes mark distinct pieces of scientific writing. Connections
2565 show references between documents.",
2566 "Nodes denote individual research articles. Links encode the
2567 act of citing prior work.",
2568 "Each node holds information about a published study. Links
2569 capture citation relationships.",
2570 "Nodes correspond to academic papers. Connections show
2571 references between documents.",
2572 "Each entry represents a scientific document. Links capture
2573 citation relationships.",
2574 "Each node holds information about a published study. Edges
2575 denote academic referencing paths.",
2576 "Nodes denote individual research articles. Edges reflect
2577 referenced literature connections.",
2578 "Each item represents a scholarly work. Edges reflect
2579 referenced literature connections.",
2580 "Each node holds information about a published study.
2581 Connections show references between documents.",
2582 "Each entry represents a scientific document. Edges denote
2583 academic referencing paths.",
2584 "Nodes symbolize separate entries in a publication index.
2585 Connections arise from inter-paper citations.",
2586 "Nodes correspond to academic papers. Links encode the act of
2587 citing prior work.",
2588 "Nodes capture metadata about research manuscripts.
2589 Connections arise from inter-paper citations.",
2590 "Each item represents a scholarly work. Edges indicate one
2591 paper citing another.",
2592 "Each node holds information about a published study. Edges
2593 indicate one paper citing another.",
2594 "Nodes correspond to academic papers. Links capture citation
2595 relationships.",

2592 "Each entry represents a scientific document. Edges denote
2593 academic referencing paths.",
2594 "Nodes denote individual research articles. Connections arise
2595 from inter-paper citations.",
2596 "Nodes capture metadata about research manuscripts. Links
2597 encode the act of citing prior work.",
2598 "Nodes symbolize separate entries in a publication index.
2599 Edges tie together documents through citations.",
2600 "Each item represents a scholarly work. Edges point to
2601 citation-based links.",
2602 "Every node is tied to a research publication. Edges denote
2603 academic referencing paths.",
2604 "Each node holds information about a published study. Edges
2605 point to citation-based links.",
2606 "Nodes denote individual research articles. Links capture
2607 citation relationships.",
2608 "Each item represents a scholarly work. Connections arise from
2609 inter-paper citations.",
2610 "Nodes symbolize separate entries in a publication index.
2611 Edges indicate one paper citing another.",
2612 "Each entry represents a scientific document. Edges reflect
2613 referenced literature connections.",
2614 "Nodes correspond to academic papers. Edges denote academic
2615 referencing paths.",
2616 "Nodes capture metadata about research manuscripts.
2617 Connections arise from inter-paper citations.",
2618 "Each item represents a scholarly work. Links capture citation
2619 relationships.",
2620 "Every node is tied to a research publication. Connections
2621 show references between documents.",
2622 "Nodes denote individual research articles. Connections show
2623 references between documents.",
2624 "Nodes symbolize separate entries in a publication index.
2625 Connections arise from inter-paper citations.",
2626 "Nodes capture metadata about research manuscripts. Edges tie
2627 together documents through citations.",
2628 "Each node holds information about a published study. Edges
2629 tie together documents through citations.",
2630 "Each entry represents a scientific document. Connections
2631 arise from inter-paper citations.",
2632 "Each node holds information about a published study.
2633 Connections arise from inter-paper citations.",
2634 "Nodes symbolize separate entries in a publication index.
2635 Edges reflect referenced literature connections.",
2636 "Nodes correspond to academic papers. Edges reflect referenced
2637 literature connections.",
2638 "Each item represents a scholarly work. Links encode the act
2639 of citing prior work.",
2640 "Nodes denote individual research articles. Connections arise
2641 from inter-paper citations.",
2642 "Each node holds information about a published study.
2643 Connections show references between documents.",
2644 "Edges tie together documents through citations. These nodes
2645 denote individual research articles.",
2646 "Each connection maps a citation trace. Every node holds
2647 information about a published study.",
2648 "Edges reflect referenced literature connections. Every node
2649 holds information about a published study.",

2646 "Connections arise from inter-paper citations. These nodes
2647 denote individual research articles.",
2648 "Connections arise from inter-paper citations. Every node is
2649 tied to a research publication.",
2650 "Connections show references between documents. These elements
2651 reflect authored academic contributions.",
2652 "Edges tie together documents through citations. The nodes
2653 mark distinct pieces of scientific writing.",
2654 "Links capture citation relationships. Nodes capture metadata
2655 about research manuscripts.",
2656 "Connections show references between documents. Each entry
2657 represents a scientific document.",
2658 "Links capture citation relationships. Each entry represents a
2659 scientific document.",
2660 "Connections arise from inter-paper citations. These nodes
2661 denote individual research articles.",
2662 "Edges reflect referenced literature connections. Every node
2663 is tied to a research publication.",
2664 "Edges point to citation-based links. The nodes mark distinct
2665 pieces of scientific writing.",
2666 "Edges denote academic referencing paths. Each item represents
2667 a scholarly work.",
2668 "Links capture citation relationships. These elements reflect
2669 authored academic contributions.",
2670 "Edges point to citation-based links. Nodes symbolize separate
2671 entries in a publication index.",
2672 "Connections show references between documents. These nodes
2673 denote individual research articles.",
2674 "Links capture citation relationships. Nodes symbolize
2675 separate entries in a publication index.",
2676 "Links encode the act of citing prior work. Nodes symbolize
2677 separate entries in a publication index.",
2678 "Edges indicate one paper citing another. Nodes symbolize
2679 separate entries in a publication index.",
2680 "Links capture citation relationships. Nodes capture metadata
2681 about research manuscripts.",
2682 "Edges reflect referenced literature connections. Nodes
2683 symbolize separate entries in a publication index.",
2684 "Links encode the act of citing prior work. Each item
2685 represents a scholarly work.",
2686 "Links capture citation relationships. Nodes symbolize
2687 separate entries in a publication index.",
2688 "Links capture citation relationships. Every node holds
2689 information about a published study.",
2690 "Links capture citation relationships. Every node is tied to a
2691 research publication.",
2692 "Connections arise from inter-paper citations. These nodes
2693 denote individual research articles.",
2694 "Edges indicate one paper citing another. Nodes capture
2695 metadata about research manuscripts.",
2696 "Links encode the act of citing prior work. The nodes
2697 correspond to academic papers.",
2698 "Links encode the act of citing prior work. These elements
2699 reflect authored academic contributions.",
"Edges indicate one paper citing another. Every node is tied
to a research publication.",
"Links capture citation relationships. The nodes correspond to
academic papers.",

2700 "Each connection maps a citation trace. Every node is tied to
2701 a research publication.",
2702 "Edges tie together documents through citations. Every node
2703 holds information about a published study.",
2704 "Connections show references between documents. The nodes mark
2705 distinct pieces of scientific writing.",
2706 "Edges denote academic referencing paths. Each item represents
2707 a scholarly work.",
2708 "Links encode the act of citing prior work. Each entry
2709 represents a scientific document.",
2710 "Edges denote academic referencing paths. These nodes denote
2711 individual research articles.",
2712 "Links capture citation relationships. The nodes correspond to
2713 academic papers.",
2714 "Links encode the act of citing prior work. The nodes
2715 correspond to academic papers.",
2716 "Each connection maps a citation trace. These nodes denote
2717 individual research articles.",
2718 "Links encode the act of citing prior work. The nodes
2719 correspond to academic papers.",
2720 "Links capture citation relationships. Each entry represents a
2721 scientific document.",
2722 "Edges point to citation-based links. Each entry represents a
2723 scientific document.",
2724 "Edges point to citation-based links. Nodes capture metadata
2725 about research manuscripts.",
2726 "Each connection maps a citation trace. Every node holds
2727 information about a published study.",
2728 "Connections show references between documents. Every node is
2729 tied to a research publication.",
2730 "Edges tie together documents through citations. Each item
2731 represents a scholarly work.",
2732 "Connections arise from inter-paper citations. These elements
2733 reflect authored academic contributions.",
2734 "Each connection maps a citation trace. These elements reflect
2735 authored academic contributions.",
2736 "The graph represents academic documents linked through
2737 references. Every node is tied to a research publication.
2738 Edges point to citation-based links.",
2739 "The structure is a scholarly graph of interconnected
2740 research. Each item represents a scholarly work. Edges point
2741 to citation-based links.",
2742 "The graph models a community of papers and their references.
2743 Each item represents a scholarly work. Each connection maps a
2744 citation trace.",
2745 "The structure is a scholarly graph of interconnected
2746 research. These elements reflect authored academic
2747 contributions. Links encode the act of citing prior work.",
2748 "This is a directed graph capturing citation flows among
2749 articles. Each entry represents a scientific document. Edges
2750 tie together documents through citations.",
2751 "This is a network of publications with citation
2752 relationships. The nodes correspond to academic papers.
2753 Connections show references between documents.",
2754 "A citation web connects scholarly articles in this dataset.
2755 These nodes denote individual research articles. Edges tie
2756 together documents through citations.",
2757

2754 "This dataset builds a graph of paper citations. Every node
2755 holds information about a published study. Each connection
2756 maps a citation trace.",
2757 "The graph models a community of papers and their references.
2758 Every node is tied to a research publication. Links capture
2759 citation relationships.",
2760 "A citation web connects scholarly articles in this dataset.
2761 Every node holds information about a published study. Edges
2762 indicate one paper citing another.",
2763 "The graph represents academic documents linked through
2764 references. These nodes denote individual research articles.
2765 Connections arise from inter-paper citations.",
2766 "This dataset forms a citation network of scientific
2767 publications. Every node is tied to a research publication.
2768 Each connection maps a citation trace.",
2769 "The graph models a community of papers and their references.
2770 Nodes symbolize separate entries in a publication index. Edges
2771 reflect referenced literature connections.",
2772 "The structure is a scholarly graph of interconnected
2773 research. Each entry represents a scientific document.
2774 Connections arise from inter-paper citations.",
2775 "A network where nodes are papers and edges are citations.
2776 These nodes denote individual research articles. Connections
2777 show references between documents.",
2778 "A citation web connects scholarly articles in this dataset.
2779 Each item represents a scholarly work. Connections arise from
2780 inter-paper citations.",
2781 "This dataset forms a citation network of scientific
2782 publications. Every node is tied to a research publication.
2783 Edges denote academic referencing paths.",
2784 "A network where nodes are papers and edges are citations.
2785 These elements reflect authored academic contributions. Edges
2786 denote academic referencing paths.",
2787 "This structure captures how academic works cite each other.
2788 Each entry represents a scientific document. Connections show
2789 references between documents.",
2790 "A network where nodes are papers and edges are citations.
2791 Every node holds information about a published study. Edges
2792 reflect referenced literature connections.",
2793 "This is a network of publications with citation
2794 relationships. These elements reflect authored academic
2795 contributions. Edges indicate one paper citing another.",
2796 "The structure is a scholarly graph of interconnected
2797 research. Every node holds information about a published
2798 study. Edges tie together documents through citations.",
2799 "The graph represents academic documents linked through
2800 references. The nodes correspond to academic papers.
2801 Connections show references between documents.",
2802 "A citation web connects scholarly articles in this dataset.
2803 Every node is tied to a research publication. Edges reflect
2804 referenced literature connections.",
2805 "This dataset builds a graph of paper citations. Nodes
2806 symbolize separate entries in a publication index. Edges
2807 reflect referenced literature connections.",
2808 "This dataset builds a graph of paper citations. The nodes
2809 correspond to academic papers. Connections arise from
2810 inter-paper citations.",
2811

2808 "The structure is a scholarly graph of interconnected
2809 research. Each item represents a scholarly work. Edges denote
2810 academic referencing paths.",
2811 "A citation web connects scholarly articles in this dataset.
2812 The nodes correspond to academic papers. Edges indicate one
2813 paper citing another.",
2814 "This is a directed graph capturing citation flows among
2815 articles. The nodes correspond to academic papers. Edges point
2816 to citation-based links.",
2817 "This is a network of publications with citation
2818 relationships. Each entry represents a scientific document.
2819 Edges reflect referenced literature connections.",
2820 "The graph models a community of papers and their references.
2821 These elements reflect authored academic contributions. Edges
2822 denote academic referencing paths.",
2823 "The graph models a community of papers and their references.
2824 Nodes symbolize separate entries in a publication index. Edges
2825 tie together documents through citations.",
2826 "A citation web connects scholarly articles in this dataset.
2827 Each item represents a scholarly work. Edges point to
2828 citation-based links.",
2829 "The graph represents academic documents linked through
2830 references. Nodes capture metadata about research manuscripts.
2831 Each connection maps a citation trace.",
2832 "The graph represents academic documents linked through
2833 references. These nodes denote individual research articles.
2834 Each connection maps a citation trace.",
2835 "The graph models a community of papers and their references.
2836 Nodes symbolize separate entries in a publication index.
2837 Connections arise from inter-paper citations.",
2838 "The graph models a community of papers and their references.
2839 These elements reflect authored academic contributions. Links
2840 encode the act of citing prior work.",
2841 "The structure is a scholarly graph of interconnected
2842 research. Nodes symbolize separate entries in a publication
2843 index. Edges reflect referenced literature connections.",
2844 "This dataset builds a graph of paper citations. The nodes
2845 correspond to academic papers. Edges denote academic
2846 referencing paths.",
2847 "This structure captures how academic works cite each other.
2848 The nodes correspond to academic papers. Edges reflect
2849 referenced literature connections.",
2850 "This is a directed graph capturing citation flows among
2851 articles. Every node is tied to a research publication.
2852 Connections show references between documents.",
2853 "A citation web connects scholarly articles in this dataset.
2854 Every node is tied to a research publication. Connections
2855 arise from inter-paper citations.",
2856 "This structure captures how academic works cite each other.
2857 These nodes denote individual research articles. Edges
2858 indicate one paper citing another.",
2859 "A network where nodes are papers and edges are citations.
2860 Each item represents a scholarly work. Edges point to
2861 citation-based links.",
2862 "A network where nodes are papers and edges are citations.
2863 Every node holds information about a published study.
2864 Connections arise from inter-paper citations.",
2865

2862 "This is a directed graph capturing citation flows among
2863 articles. Each item represents a scholarly work. Edges
2864 indicate one paper citing another."
2865]

2866
2867

2868 H.5 OGBN-PRODUCTS GRAPHS DESCRIPTIONS DATASET

2869
2870 [

2871 "the network's nodes are products, and the edges connecting
2872 them show that those products are often bought in the same
2873 purchase",
2874 "in this graph, products are represented by nodes, and a
2875 shared purchase is shown by an edge between two products",
2876 "edges link products (nodes) that are purchased
2877 simultaneously, while nodes signify the products themselves",
2878 "a node stands for a product, and an edge between two nodes
2879 means those two products are frequently part of the same
2880 transaction",
2881 "products are the nodes in this network, and an edge between
2882 any two of them signifies they are commonly bought together",
2883 "this model uses nodes for products and edges to represent
2884 joint purchases of those products",
2885 "each product is a node, and an edge is drawn between two
2886 products when they are purchased in a single transaction",
2887 "nodes are individual products, and an edge connecting any two
2888 nodes shows they are bought as a pair",
2889 "if products are nodes, then a connecting edge indicates that
2890 those products were purchased at the same time",
2891 "nodes in this graph are products, and their shared purchase
2892 is symbolized by an edge linking them",
2893 "products sold are represented by nodes, and edges show which
2894 pairs of products were acquired in the same purchase",
2895 "in this system, products are nodes, and an edge exists if two
2896 products are bought together",
2897 "the nodes are products, and an edge between them denotes a
2898 co-purchase of those items",
2899 "a product is a node, and an edge between two nodes indicates
2900 a simultaneous purchase of the products",
2901 "nodes symbolize products, and an edge between them represents
2902 that they are typically purchased as a set",
2903 "products are nodes, and an edge connecting them signifies
2904 that they are frequently purchased together in one order",
2905 "this network uses nodes for products and edges to indicate
2906 that two products are found in the same transaction",
2907 "each product is a node, and an edge between two nodes
2908 signifies that the products were purchased together",
2909 "the nodes in this graph are products, and the edges represent
2910 a relationship of being bought at the same time",
2911 "products are nodes, and an edge between them means they are
2912 part of a single purchase",
2913 "nodes stand for products, while edges link any two products
2914 that were bought in a single transaction",
2915 "products are represented by nodes, and edges connecting those
2916 nodes signify they are often purchased together",
2917 "the nodes are products, and the edges show a connection based
2918 on them being bought in the same transaction",
2919 "a node represents a product, and an edge between two products
2920 shows they are purchased together",

2916 "nodes symbolize products, and edges connect those products
2917 that are found together in a single sale",
2918 "in this model, products are nodes, and edges are drawn to
2919 indicate which products are purchased together",
2920 "nodes are products sold, and an edge between two nodes
2921 indicates they are included in the same purchase",
2922 "the network's nodes are products, and edges connect products
2923 that are commonly purchased in the same transaction",
2924 "an edge between two product nodes means those items were
2925 purchased at the same time",
2926 "nodes are products, and an edge is drawn between products
2927 that are frequently purchased together",
2928 "a node represents a product, and an edge between two nodes
2929 shows they are frequently purchased together",
2930 "the nodes of this graph are products, and the edges signify
2931 that two products were bought as a pair",
2932 "products are nodes, and an edge indicates a joint purchase of
2933 those products",
2934 "a product is a node, and an edge between two nodes means they
2935 are co-purchased",
2936 "the nodes represent products, and the edges connect products
2937 that are bought together",
2938 "nodes are products, and a link between two products means
2939 they were purchased as a set",
2940 "in this representation, products are nodes, and edges between
2941 them represent simultaneous purchases",
2942 "products are the nodes, and a connecting edge signifies that
2943 they are bought in the same transaction",
2944 "nodes symbolize products, and an edge is created when those
2945 products are bought together",
2946 "the nodes are products sold, and the edges indicate which
2947 products are purchased together",
2948 "in this network, products are nodes, and an edge shows that
2949 they are co-purchased",
2950 "nodes stand for products, and edges represent a connection of
2951 being purchased together",
2952 "each product is a node, and an edge between two products
2953 shows a common purchase",
2954 "the nodes represent products, and edges are used to show
2955 products that are purchased in a single transaction",
2956 "products are nodes, and an edge between them represents their
2957 co-occurrence in a purchase",
2958 "nodes are products, and an edge between them signifies a
2959 simultaneous purchase of those items",
2960 "products are nodes, and an edge between them indicates a
2961 joint purchase",
2962 "a product is a node, and an edge means those two products are
2963 purchased together",
2964 "the network's nodes are products, and edges show which
2965 products are bought in the same order",
2966 "nodes are products sold, and edges between two products mean
2967 they are purchased at the same time",
2968 "in this structure, nodes are products, and edges represent
2969 that they were purchased together",
2970 "products are nodes, and an edge is drawn between them when
2971 they are purchased together",
2972 "the nodes represent products, and an edge connects products
2973 that are bought in the same purchase",
2974

3024 "products are nodes, and an edge shows they are bought
3025 together",
3026 "the nodes are products, and an edge indicates a simultaneous
3027 purchase",
3028 "a node is a product, and an edge means a joint purchase",
3029 "products are nodes, and an edge shows they were bought
3030 together",
3031 "the nodes are products, and an edge signifies a co-purchase",
3032 "a node is a product, and an edge means they are bought
3033 together",
3034 "products are nodes, and an edge represents a joint purchase",
3035 "the nodes are products, and edges show they are purchased
3036 together",
3037 "a node is a product, and an edge means a co-purchase",
3038 "products are nodes, and an edge indicates a joint purchase",
3039 "the nodes are products, and an edge shows they are bought
3040 together",
3041 "a node is a product, and an edge means a co-purchase",
3042 "products are nodes, and an edge indicates a co-purchase",
3043 "the nodes are products, and an edge shows they are bought
3044 together",
3045 "a node is a product, and an edge means a joint purchase",
3046 "products are nodes, and an edge represents a co-purchase",
3047 "edges between two products signify they are purchased
3048 together, while the nodes themselves represent the products
3049 being sold",
3050 "an edge between two products indicates a shared purchase, and
3051 the products themselves are represented by the network's
3052 nodes",
3053 "the network's edges show that two products are bought in the
3054 same purchase, with nodes symbolizing the products",
3055 "a connecting edge means two products were purchased
3056 simultaneously, whereas each product is represented as a
3057 node",
3058 "edges link products (nodes) that are purchased together, and
3059 each node individually represents a sold product",
3060 "the edges in this graph represent joint purchases, and the
3061 nodes are the individual products involved",
3062 "edges indicate that two products are found in the same
3063 transaction, while the nodes represent the products",
3064 "if an edge exists, the two products (nodes) it connects are
3065 purchased together in a single transaction",
3066 "a shared purchase is represented by an edge, and the products
3067 themselves are the nodes in this network",
3068 "edges are drawn between products to show they are bought
3069 together; the products themselves are the nodes",
3070 "the edges signify a co-purchase of two products, which are
3071 represented by the network's nodes",
3072 "edges between two products indicate a shared purchase, and
3073 the products sold are the nodes",
3074 "a joint purchase of two products is represented by an edge,
3075 and the products themselves are the nodes",
3076 "an edge between two nodes indicates a simultaneous purchase,
3077 while the nodes are the products",
3078 "edges show that two products are purchased together, and each
3079 product is a node in the graph",
3080 "the edges link products that are bought together, and the
3081 products are represented as nodes",

3078 "edges signify a co-purchase of the two products that they
3079 connect; those products are the nodes",
3080 "the relationship of being purchased together is represented
3081 by edges, and the products are the nodes",
3082 "edges show which products were bought at the same time, and
3083 the products themselves are the nodes",
3084 "the edges represent a co-purchase, and each product is a node
3085 in this network",
3086 "an edge connecting two products shows they were purchased
3087 together; nodes are the products sold",
3088 "a connecting edge indicates that two products were bought
3089 simultaneously, with nodes representing the products",
3090 "edges are used to show products that are purchased in a
3091 single transaction, and nodes are the products",
3092 "the edges represent products that are part of the same
3093 transaction, and the products are nodes",
3094 "edges connect products that are purchased in the same order;
3095 the nodes are the products",
3096 "the edges indicate a joint purchase of products, which are
3097 represented by the nodes",
3098 "edges represent a co-purchase of products, and the products
3099 sold are the nodes",
3100 "a connecting edge signifies a joint purchase, while the nodes
3101 represent the products",
3102 "the edges show that two products were purchased together,
3103 with the nodes being the products",
3104 "edges show a connection based on a shared purchase, and the
3105 nodes are the products",
3106 "an edge between two products means they were purchased
3107 together, and the products are nodes",
3108 "the edges signify a simultaneous purchase of the products,
3109 which are represented by the nodes",
3110 "a joint purchase is represented by an edge, while the
3111 products are nodes",
3112 "edges link products that are bought together, and nodes are
3113 the products themselves",
3114 "the edges indicate a co-purchase of products, and the
3115 products are the nodes",
3116 "an edge between two nodes means the products they represent
3117 were purchased together",
3118 "edges show a co-purchase of products, which are represented
3119 by nodes",
3120 "a connecting edge represents that two products were bought
3121 together, and nodes are the products",
3122 "the edges represent a joint purchase, and the products are
3123 nodes",
3124 "an edge indicates a simultaneous purchase, while the products
3125 are represented by nodes",
3126 "edges are drawn between products that are bought together,
3127 and the products are nodes",
3128 "the edges signify a co-purchase, with the products being the
3129 nodes",
3130 "an edge shows that two products were purchased together, and
3131 the products are the nodes",
3132 "the edges represent a joint purchase, and the nodes are the
3133 products",
3134 "a connecting edge means two products were bought together,
3135 and nodes are the products",

3186 "the edges signify a joint purchase, with the products being
3187 the nodes",
3188 "an edge indicates that two products were purchased together,
3189 and nodes are the products",
3190 "edges represent a co-purchase, and the products are the
3191 nodes",
3192 "the edges show a joint purchase, while the products are
3193 nodes",
3194 "an edge means that two products were bought together, and the
3195 nodes are the products",
3196 "edges represent a co-purchase, and the products are nodes",
3197 "the edges signify a joint purchase, and the nodes are the
3198 products",
3199 "an edge indicates that two products were purchased together,
3200 and nodes are the products",
3201 "edges represent a co-purchase, and the products are the
3202 nodes",
3203 "the edges show a joint purchase, while the products are
3204 nodes",
3205 "an edge means two products were bought together, and the
3206 products are nodes",
3207 "edges represent a co-purchase, and the nodes are the
3208 products",
3209 "the edges signify a joint purchase, with the products being
3210 the nodes",
3211 "an edge indicates that two products were purchased together,
3212 and nodes are the products",
3213 "edges represent a co-purchase, and the products are the
3214 nodes",
3215 "the edges show a joint purchase, while the products are
3216 nodes",
3217 "an edge means that two products were bought together, and the
3218 nodes are the products",
3219 "edges represent a co-purchase, and the products are nodes",
3220 "the edges signify a joint purchase, and the nodes are the
3221 products",
3222 "an edge indicates that two products were purchased together,
3223 and nodes are the products",
3224 "edges represent a co-purchase, and the products are the
3225 nodes",
3226 "the edges show a joint purchase, while the products are
3227 nodes",
3228 "nodes in the graph are the products sold, and an edge between
3229 two signifies they are purchased together"

]

3230
3231
3232
3233
3234
3235
3236
3237
3238
3239