DGP: A Dual-Granularity Prompting Framework for Fraud Detection with Graph-Enhanced LLMs

Yuan Li[1*], **Jun Hu**[1†], **Bryan Hooi**[1], **Bingsheng He**[1], **Cheng Chen**[2]
[1]National University of Singapore
[2]ByteDance Inc.
li.yuan@u.nus.edu, {jun.hu, dcsbhk, dcsheb}@nus.edu.sg, chencheng.sg@bytedance.com

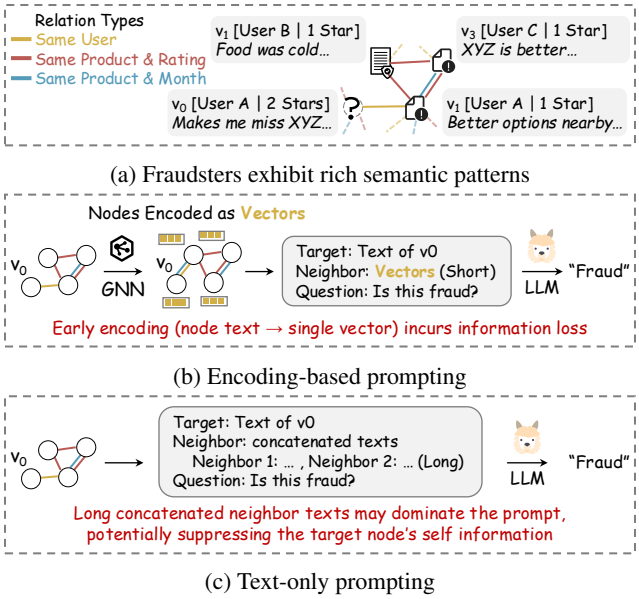
Abstract

Real-world fraud detection applications benefit from graph learning techniques that jointly exploit node features—often rich in textual data—and graph structural information. Recently, Graph-Enhanced LLMs have emerged as a promising graph learning approach that converts graph information into prompts, exploiting LLMs' ability to reason over both textual and structural information. Among them, text-only prompting, which converts graph information into prompts consisting solely of text tokens, offers a solution that relies only on LLM tuning without requiring additional graph-specific encoders. However, text-only prompting struggles on heterogeneous fraud-detection graphs: multi-hop relations expand exponentially with each additional hop, leading to rapidly growing neighborhoods associated with dense textual information. These neighborhoods may overwhelm the model with long, irrelevant content in the prompt and suppress key signals from the target node, thereby degrading performance. To address this challenge, we propose Dual Granularity Prompting (DGP), which mitigates information overload by preserving fine-grained textual details for the target node while summarizing neighbor information into coarse-grained text prompts. DGP introduces tailored summarization strategies for different data modalities—bi-level semantic abstraction for textual fields and statistical aggregation for numerical features—enabling effective compression of verbose neighbor content into concise, informative prompts. Experiments across public and industry datasets demonstrate that DGP operates within a manageable token budget while improving fraud detection performance by up to 6.8% (AUPRC) over state-of-the-art methods, showing the potential of Graph-Enhanced LLMs for fraud detection.

Code — https://github.com/Xtra-Computing/DGP

Introduction

Graph-based fraud detection has emerged as a critical research direction, driven by its effectiveness in capturing the complex relational patterns inherent in real-world data (Xu et al. 2024; Akoglu, Tong, and Koutra 2015; Rayana and Akoglu 2015). The intricate structural properties of graphs, combined with the rich semantic and numerical information on nodes, present unique opportunities and challenges for effectively identifying fraudulent entities. Real-world applications such as anomaly detection in social networks (Chen et al. 2024; Sharma et al. 2018), fake account identification (Li et al. 2022; Hooi et al. 2017), and the detection of malicious user-generated content (Rayana and Akoglu 2015) benefit from advanced graph learning techniques.



(a) Fraudsters exhibit rich semantic patterns

(b) Encoding-based prompting

(c) Text-only prompting

Figure 1: Graph-to-prompt methods for fraud detection.

Graph-Enhanced LLMs for Fraud Detection. In recent years, various Graph Neural Networks (GNNs) have been proposed for graph-based fraud detection, achieving notable success by leveraging neighborhood information and structural patterns to enhance detection accuracy (Duan et al. 2024; Li et al. 2024a). More recently, graph-enhanced Large Language Models (LLMs) have emerged as a promising alternative for graph-based fraud detection tasks, leveraging their generalizable language capabilities and demonstrating competitive performance across a range of tasks (Tang et al. 2024a; Liu et al. 2024b). These approaches have shown potential in analyzing the rich semantics associated with

[*]Work done during internship at ByteDance.
[†]Corresponding author.

fraudulent nodes, as well as the diverse relationships among them (as illustrated in Figure 1a), by exploiting the semantic nuances within the graph (Tang et al. 2024a). We distinguish these methods from LLM-enhanced GNNs such as TAPE (He et al. 2024), FLAG (Yang et al. 2025), and MLED (Huang and Wang 2025), which incorporate LLM-encoded features and train GNNs for classification. In this work, we focus on leveraging graph-enhanced LLMs as classifiers to explore their potential in graph-based fraud detection, emphasizing their flexibility in handling complex semantics and diverse queries.

To bridge the gap between graph-structured data and LLMs, graph-enhanced LLMs transform graph data into textual prompts, i.e., *graph-to-prompt*, to naturally integrate both graph structure and semantics into LLMs (Fatemi, Halcrow, and Perozzi 2023; Ye et al. 2024). Two major graph-to-prompt strategies, as depicted in Figures 1b and 1c, have been developed in recent literature: (1) Encoding-based prompting, exemplified by approaches such as GraphGPT (Tang et al. 2024a) and HiGPT (Tang et al. 2024b), encodes nodes into compact vectors and subsequently feeds them into an LLM. These methods substantially reduce prompt length via node encoding, but suffer from early vectorization, leading to **information loss** due to reduced semantic-level interactions (Li et al. 2023b). In contrast, (2) text-only prompting (Wang et al. 2023; Fatemi, Halcrow, and Perozzi 2023; Zhu et al. 2025) preserves detailed semantic interactions by concatenating neighbor texts into the prompt. However, these methods inherently suffer from excessive prompt length, leading to **distraction from crucial content** due to information overload (Li et al. 2024b). For example, in industrial scenarios, each neighboring node may be associated with over 1,500 tokens, resulting in a 2-hop neighborhood containing up to 2 million tokens. This poses significant challenges for incorporating dense textual information into fraud detection models.

In this work, we propose **D**ual **G**ranularity **P**rompting (DGP), a novel text-only prompting framework that leverages the rich semantics on graphs while addressing the challenge of excessive prompt length. To reduce information loss incurred by early-stage encoding, DGP selectively preserves fine-grained text for the target node while summarizing neighbors retrieved from different metapaths into compact, coarse-grained texts. For textual features, we employ bi-level semantic summarization to reduce prompt length, further applying metapath trimming to incorporate semantically and structurally relevant neighbors for each metapath. For numerical features, we adopt precise numerical summarization to retain key insights. As illustrated in Figure 2, our approach achieves an impressive balance between token usage and performance. Compared to prior state-of-the-art methods, DGP operates with a manageable prompt length while improving fraud detection performance by up to 6.8% (AUPRC), demonstrating the effectiveness of our dual-granularity design with manageable token budgets.

The key contribution of this work is three-fold:

- We propose DGP, a novel graph prompting framework that integrates fine-grained textual details for target nodes with coarse-grained semantic summaries for their neighbors, thereby overcoming limitations faced by existing graph-to-prompt methods.
- We introduce specialized summarization strategies for compressing neighborhoods associated with textual and numerical features into concise, semantically meaningful prompts tailored for LLM processing.
- Extensive experiments on public and industry datasets demonstrate the superior empirical performance of DGP, achieving manageable prompt lengths while improving fraud detection performance by up to 6.8% in AUPRC compared to state-of-the-art approaches.

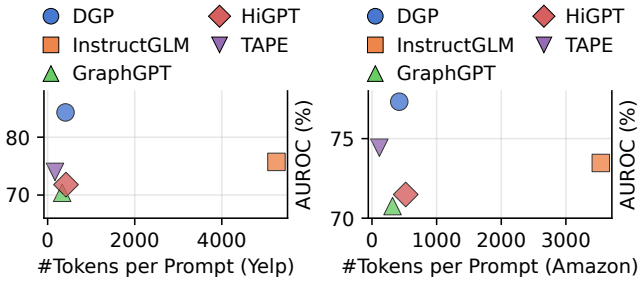


Figure 2: Fraud detection performance (↑) vs. token usage per prompt (↓) across different methods and datasets. Our proposed method, DGP, achieves top performance with moderate token consumption, demonstrating a notable balance between token usage and performance.

Related Work

Graph Neural Networks for Fraud Detection

Graph neural networks (GNNs) have become the dominant approach for fraud detection by modeling relational patterns in graphs (Akoglu, Tong, and Koutra 2015; Rayana and Akoglu 2015; Duan et al. 2024; Li et al. 2024a; Qiao et al. 2025; Niu et al. 2025). Classic models such as GCN (Kipf and Welling 2017) and GAT (Veličković et al. 2018) have inspired many variants targeting specific challenges, including camouflage (Dou et al. 2020), heterophily (Zhuo et al. 2024), limited supervision (Chen et al. 2024; Li, Yu, and Luo 2025), barely-supervised learning (Yu, Liu, and Luo 2024)), and context inconsistency (Li et al. 2023a). However, most GNN-based approaches underutilize the fine-grained textual semantics widely available in real-world graphs, which our method explicitly addresses.

Integrating LLMs with Graphs

Recent advances in integrating LLMs with graph data can be broadly classified into *graph-enhanced LLMs* and *LLM-enhanced GNNs*. Graph-enhanced LLMs primarily adopt graph-to-prompt strategies, which can be divided into encoding-based prompting and text-only prompting. Encoding-based prompting (Tang et al. 2024a,b) compresses graph features for LLM input, potentially resulting in semantic loss. Specifically, GraphGPT (Tang et al. 2024a) aligns LLMs with graph structural information via dual-stage instruction tuning. HiGPT (Tang et al. 2024b) extends

instruction tuning to heterogeneous graphs. In contrast, text-only prompting (Fatemi, Halcrow, and Perozzi 2023; Ye et al. 2024; Zhu et al. 2025) concatenates the texts of neighboring nodes as input to LLMs, which may lead to excessively long prompts and distract from crucial information. For example, InstructGLM (Ye et al. 2024) frames graph tasks as natural-language instructions.

Another line of work, LLM-enhanced GNNs (He et al. 2024; Huang and Wang 2025; Yang et al. 2025), integrates LLM-encoded features into GNN classifiers. For example, TAPE (He et al. 2024) uses LLM-generated explanations as auxiliary features for downstream GNNs, and FLAG (Yang et al. 2025) leverages discriminative text extraction to address neighborhood camouflage in fraud detection.

In this paper, we explore graph-enhanced LLMs, where LLMs operate on graph-structured information and serve as the classifier. This design allows us to explore the capabilities of LLMs for graph-based fraud detection.

Preliminaries

In this section, we formalize the fraud detection problem on heterogeneous graphs and define metapaths.

Graph-based Fraud Detection

Given a heterogeneous graph $G = \{V, E, \mathcal{R}, \mathcal{X}\}$, where V denotes a set of N nodes, $E \subseteq V \times V \times \mathcal{R}$ is the set of typed edges, $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$ is the set of edge relation types, and $\mathcal{X} = \{x_v\}_{v \in V}$ represents the node features of mixed types. Each node $v \in V$ is associated with a feature tuple $x_v = (x_v^{\text{text}}, x_v^{\text{num}})$, where x_v^{text} denotes raw textual content (e.g., user-written reviews), and $x_v^{\text{num}} \in \mathbb{R}^d$ stacks all numeric or one-hot categorical features (e.g., a rating ranging from 1 to 5 stars).

We focus on the task of node-level fraud detection. Let $y_v \in \{0, 1\}$ be a binary label indicating whether node v is fraudulent (1) or benign (0). The objective is to learn a function $f : V \longrightarrow \{0, 1\}$ that minimizes the empirical risk:

$$\mathcal{L} = \frac{1}{|V_{\text{train}}|} \sum_{v \in V_{\text{train}}} \ell\big(f(v), y_v\big) \tag{1}$$

where $\ell(\cdot, \cdot)$ is the binary cross-entropy loss, and $V_{\text{train}} \subset V$ denotes the labeled training nodes. At inference time, f is applied to each unseen node to predict its label.

Metapaths on Heterogeneous Graphs

For each relation $r \in \mathcal{R}$, we define $A_r \in \{0, 1\}^{N \times N}$ as the typed adjacency matrix, where entry $(A_r)_{uv} = 1$ if $(u, v, r) \in E$. A metapath (Sun et al. 2011) is a finite sequence of relations:

$$P = r_1 \circ r_2 \circ \cdots \circ r_L \tag{2}$$

which describes a composite semantic, e.g., $Review \rightarrow User \rightarrow Review$. The metapath-specific adjacency matrix is computed as:

$$A_P = A_{r_1} A_{r_2} \cdots A_{r_L} \tag{3}$$

and the metapath-specific neighborhood of v is defined as:

$$\mathcal{N}_P(v) = \{u \in V \mid (A_P)_{vu} > 0\} \tag{4}$$

Methodology

This section details the component design of the dual-granularity prompting framework.

Dual Granularity Prompting

Effectively bridging graph-structured data with large language models requires a careful balance between semantic richness and manageable prompt length. Existing approaches tend to fall short: encoding-based prompting compresses neighborhood information at the expense of crucial semantic cues, while text-only prompting preserves detail but quickly overwhelms LLMs with excessive prompt lengths. Inspired by the selective granularity strategies in recent GNN work such as RpHGNN (Hu, Hooi, and He 2024), which demonstrate the benefits of retaining fine-grained target node features while abstracting neighborhood context, we propose Dual Granularity Prompting (DGP) for graph-based fraud detection. DGP preserves fine-grained textual details for the target node and compresses neighbor information into concise, high-level summaries—striking a practical balance between informativeness and token budget.

As depicted in Figure 3, the DGP framework is composed of three core modules: (i) node-level summarization to distill the essence of each node's raw text, (ii) diffusion-based metapath trimming to preserve the most structurally and semantically relevant neighbors along each metapath, and (iii) metapath-level summarization to further aggregate both textual and numerical features. The resulting dual-granularity prompts enable LLMs to effectively process complex graph data, capturing essential fraud-related signals.

Textual Summarization

We detail the bi-level textual summarization process.

Node-level Summarization A significant obstacle in leveraging large language models (LLMs) for graph-based fraud detection is the sheer volume of textual data associated with nodes, particularly when considering multi-hop neighbors. To effectively address this issue, we first condense the textual description of each node into a concise yet representative intrinsic summary. Formally, given the raw textual feature x_v^{text} of node v, we generate a summarized text s_v:

$$s_v = \text{Summarize}(x_v^{\text{text}}; B_{\text{node}}) \tag{5}$$

where B_{node} denotes the token budget per node. To avoid hand-crafting domain-specific prompts, we adopt a task-agnostic summarization approach that condenses text within a fixed token budget. Although some detail may be lost, the resulting summaries remain effective for downstream metapath extraction and reasoning. In contrast, we empirically observe that task-specific prompts may underperform in the absence of dataset-specific expertise, as they can misguide the model and degrade summarization quality.

Diffusion-based Metapath Trimming Metapaths capture composite semantics in heterogeneous graphs by connecting nodes through meaningful multi-hop relational sequences.

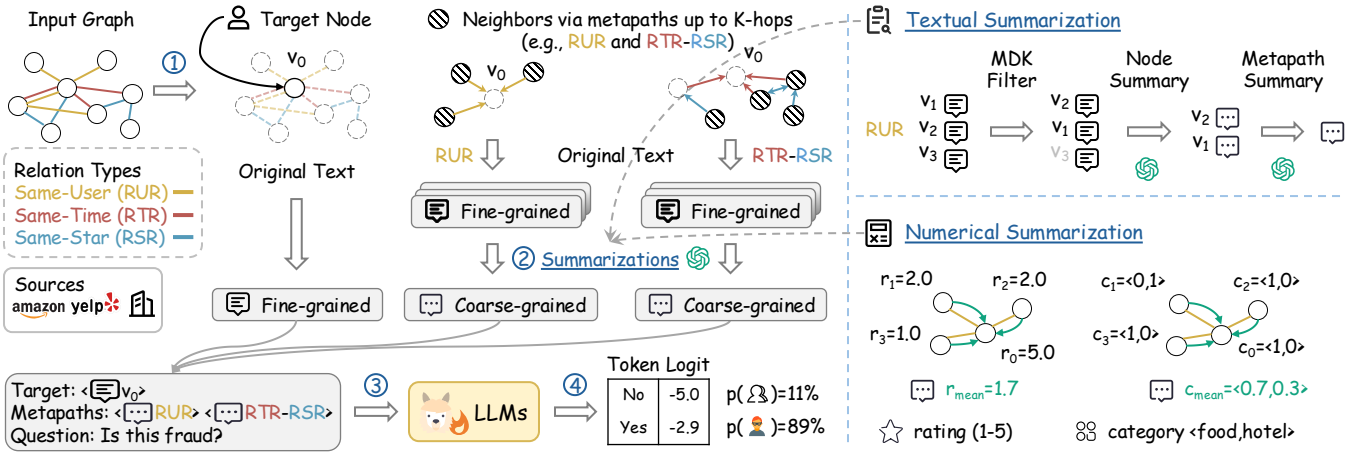


Figure 3: Overview of the proposed DGP framework.

They enable the construction of rich, type-aware neighborhoods that reflect diverse semantic views. Although node-level summarization helps reduce redundancy, directly aggregating information from all neighbors across multiple metapaths remains computationally prohibitive and prone to semantic noise. To capture the local fraud-related context of a target node v, we apply structure- and semantics-aware metapath trimming guided by the Markov Diffusion Kernel (MDK) (Fouss et al. 2012; Zhu and Koniusz 2021).

For each metapath P, we form the row-stochastic transition matrix $\mathbf{T}_P = \mathbf{D}_P^{-1}\mathbf{A}_P$ from the metapath-specific adjacency matrix $\mathbf{A}_P$ and degree matrix $\mathbf{D}_P = \mathrm{diag}(\mathbf{A}_P\mathbf{1})$. Averaging the first K random-walk powers, i.e., K-hops, yields the Markov diffusion operator:

$$\mathbf{Z}_P(K) = \frac{1}{K}\sum_{k=0}^{K}\mathbf{T}_P^k \tag{6}$$

Let $\mathbf{X} = \mathrm{emb}(\mathbf{X}^{\text{text}}) \oplus \mathbf{X}^{\text{num}} \in \mathbb{R}^{N\times(d_{\text{LM}}+d)}$ denote the node embeddings, where text is embedded using DeBERTa (He, Gao, and Chen 2021). We propagate $\mathbf{X}$ via $\mathbf{Z}_P(K)$ to obtain structure-aware semantic embeddings:

$$\mathbf{h}_i^{(P)}(K) = [\mathbf{Z}_P(K)\mathbf{X}]_{i:} \tag{7}$$

where $\mathbf{h}_i^{(P)}(K)$ is the diffused embedding of node i under metapath P, corresponding to the i-th row of the matrix $\mathbf{Z}_P(K)\mathbf{X}$. The joint diffusion distance between nodes u and v is defined as:

$$\delta_K^{(P)}(u,v) = \left\|\mathbf{h}_u^{(P)}(K) - \mathbf{h}_v^{(P)}(K)\right\|_2 \tag{8}$$

which measures how similarly u and v diffuse information along metapath P.

To reduce the semantic noise and focus on relevant context, we retain the top-M nearest neighbors of the target node v based on diffusion distance:

$$\widetilde{\mathcal{N}}_P(v) = \underset{u\in\mathcal{N}_P(v)}{\mathrm{TopM}}\left(-\delta_K^{(P)}(u,v)\right) \tag{9}$$

This results in a pruned, structure- and semantics-aware neighbor set suitable for downstream fraud detection tasks.

Metapath Summarization We aggregate the node-level summaries of the pruned neighbors under each metapath P into a metapath summary $S_P(v)$:

$$S_P(v) = \mathrm{Summarize}\left(\oplus_{u\in\widetilde{\mathcal{N}}_P(v)} s_u;\ B_{\text{meta}}\right) \tag{10}$$

where $\oplus$ denotes concatenation and B_{meta} is the token budget per metapath summary. This summarization further reduces redundancy by synthesizing a concise and informative metapath-level textual representation.

Numerical Summarization

Unlike textual data, numerical and categorical features often encode precise signals critical for fraud detection. To retain this information, we perform mean aggregation along each metapath. For a target node v and metapath P, the aggregated representation is defined as:

$$a_P(v) = \frac{1}{|\widetilde{\mathcal{N}}_P(v)|}\sum_{u\in\widetilde{\mathcal{N}}_P(v)} x_u^{\text{num}} \tag{11}$$

where x_u^{num} denotes either a real-valued numerical feature or a categorical vector encoded as one-hot or multi-hot. This formulation allows us to summarize the distributional properties of both continuous and discrete structured features, providing a complementary signal to the textual summaries.

Fraud Detection with DGP

To incorporate both textual and numerical features, we construct structured prompts:

$$\mathrm{prompt}(v) = x_v^{\text{text}} \oplus \left[\bigoplus_{P\in\mathcal{P}_K}\left(S_P(v)\oplus a_P(v)\right)\right] \tag{12}$$

where $\mathcal{P}_K$ denotes the set of metapath types up to K hops. We finetune the LLM on labeled nodes by minimizing the cross-entropy loss over the first generated token:

$$\mathcal{L} = -\frac{1}{|V_{\text{train}}|}\sum_{v\in V_{\text{train}}}\log p_\theta(y_v \mid \mathrm{prompt}(v)) \tag{13}$$

Dataset	Node Type	Textual	Numerical	# Nodes	# Edges	# Edge Types	# Frauds	# Train / Val / Test
YelpReviews	Service Review	✓	✓	67,395	17,486,608	3	8,919	1,348 / 1,348 / 13,479
AmazonVideo	Product Review	✓	✓	37,126	9,883,406	3	4,379	1,299 / 1,299 / 7,425
E-Commerce	Shop Profile	✓	✓	182,043	27,196,608	9	3,256	1,309 / 1,309 / 3,928
LifeService	Shop Profile	✓	✓	12,868	82,912	5	2,868	1,287 / 1,287 / 2,574

Table 1: Overview of the datasets.

where $y_v \in \{\texttt{Yes}, \texttt{No}\}$ denotes the correct label, and $p_\theta(y_v \mid \text{prompt}(v))$ represents the token probability output by the LLM.

During inference, we apply softmax over the logits of the first generated token to compute the fraud probability:

$$p_v = \frac{\exp(\text{logit}_{\texttt{Yes}})}{\exp(\text{logit}_{\texttt{Yes}}) + \exp(\text{logit}_{\texttt{No}})} \tag{14}$$

where $\text{logit}_{\texttt{Yes}}$ and $\text{logit}_{\texttt{No}}$ are the pre-softmax scores assigned by the LLM to the tokens `Yes` and `No`, respectively. The fraud probability p_v is interpreted as the model's confidence that node v is fraudulent.

Complexity Analysis of DGP

Token Consumption For a single target node, let L denote the average token length of a node's text, D the average out-degree, R the number of relation types, B the summarization budget, K the number of hops, and M the metapath neighbor truncation. The prompt length for a full-neighbor approach is $\frac{D^{K+1}-1}{D-1}L$. Similarly, a fully-vectorized prompt consumes $L + \frac{D^{K+1}-D}{D-1}$ tokens, condensing neighbor texts into vectors. For DGP, the bi-level summarization prompts use $L + \frac{R^{K+1}-R}{R-1}MB$ tokens, which scale with R^K instead of D^K. The final prompt consumes $L + \frac{R^{K+1}-R}{R-1}B$ tokens for fraud detection on the target node.

Since R, K, and M are typically small in practice, DGP achieves significant token savings compared to full-neighbor methods. We note that D can be much larger than R in real-world heterogeneous graphs (e.g., $D = 133$ on the Amazon dataset), resulting in much longer full-neighbor prompts. Meanwhile, the average node text length L continues to grow in modern web-scale datasets (e.g., $L > 1{,}500$ on industry datasets), further amplifying the advantage of DGP's bi-level summarization design.

Time Complexity The training phase consists of two frozen-LLM summarization passes and one fine-tuning loop. Processing all N nodes with both node- and metapath-level summaries costs $\mathcal{O}\big((L+B)^2N\big)$ and $\mathcal{O}\big((\frac{R^{K+1}-R}{R-1}MB)^2N\big)$, respectively. Finetuning on $\mathcal{O}(N)$ labeled nodes for E epochs, each with a sequence length of $L + \frac{R^{K+1}-R}{R-1}B$, costs $\mathcal{O}\big((L + \frac{R^{K+1}-R}{R-1}B)^2EN\big)$.

During inference, the bi-level summaries are cached, so each of the N nodes requires only one forward pass of length $L + \frac{R^{K+1}-R}{R-1}B$, giving $\mathcal{O}\big((L + \frac{R^{K+1}-R}{R-1}B)^2N\big)$. Thus, the overall inference complexity scales linearly with the number of nodes N and remains unaffected by large out-degree D, highlighting DGP's practicality in real-world applications. Importantly, the ratio between the target node's input length L and the neighbor input length $\frac{R^{K+1}-R}{R-1}B$ can be flexibly controlled via hyperparameters K and B. This design mitigates the risk of neighbor information dominating the prompt and reduces computation on large multi-hop neighborhoods, ensuring that the model remains focused on the target node while maintaining practicality.

Attention Dilution in Fraud Detection

We provide theoretical insights into how excessive neighbor information can overwhelm fraud signals and demonstrate how our approach mitigates this issue. Let the input to the Transformer-based LLM backbone (Vaswani et al. 2017) consist of L tokens representing the target node and mn_K tokens representing its K-hop neighbors, where the total sequence length is $T_K = L + mn_K$. We denote the contribution from each neighbor node as m tokens and define the size of the K-hop neighborhood as $n_K = \frac{D^{K+1}-D}{D-1}$, which grows exponentially with the average out-degree D. We assume that the target node is a fraudulent node we aim to detect. Suppose the global fraud ratio is $p \ll 1$, which is typical in real-world graphs. As the number of neighbors increases, the fraction of fraud-related tokens in the prompt (including the target node itself) is given by:

$$r = \frac{L + p\,mn_K}{L + mn_K} = p + \frac{L(1-p)}{L + mn_K} \le p + \frac{L(1-p)}{mn_K} \tag{15}$$

which gradually decreases from 1 to p. Notably, the softmax attention mechanism (Vaswani et al. 2017) is given by:

$$\alpha_i = \frac{\exp(q^\top k_i/\sqrt{d})}{\sum_{j=1}^{T_K}\exp(q^\top k_j/\sqrt{d})}, \tag{16}$$

where d denotes the model dimension, q is the query vector (i.e., the last input token), and k represents the key vectors. Under the assumption that token similarities are roughly uniform, the expected attention mass assigned to fraud-related tokens is r, which rapidly diminishes among the large number of benign tokens. This effect is well studied as attention dispersion or over-squashing (Liu et al. 2024a; Barbero et al. 2024; Vasylenko, Treviso, and Martins 2025), where important signals are easily overwhelmed by irrelevant context in long sequences. Our bi-level summarization alleviates this issue by controlling m and n_K, allowing the model to focus on informative fraud patterns.

Experiments

We conduct extensive experiments to evaluate DGP from three perspectives: (i) *effectiveness* — how well DGP per-

Dataset	YelpReviews			AmazonVideo			E-Commerce			LifeService		
Method	MacroF1	AUROC	AUPRC	MacroF1	AUROC	AUPRC	MacroF1	AUROC	AUPRC	MacroF1	AUROC	AUPRC
MLP	62.09±0.05	75.00±0.06	32.24±0.13	61.74±0.39	70.47±0.18	26.55±0.48	65.21±0.08	71.02±0.11	68.14±0.20	87.98±0.08	95.49±0.04	88.15±0.21
SAGE	64.64±0.69	75.59±0.89	36.75±1.99	62.23±0.08	70.11±0.37	25.93±0.54	68.66±0.54	73.85±0.77	71.84±1.21	88.42±0.34	95.42±0.26	88.09±0.75
HGT	66.53±0.57	81.49±0.50	40.04±1.64	65.55±0.59	73.07±0.70	33.41±0.66	65.07±0.75	72.05±0.82	68.42±1.17	89.28±0.30	95.82±0.25	89.90±1.16
ConsisGAD	67.33±0.05	82.12±0.21	42.11±0.16	63.92±1.89	74.07±1.83	29.73±2.42	69.58±0.50	77.10±0.36	76.40±0.40	90.55±0.28	96.98±0.12	92.85±0.24
PMP	63.76±2.87	78.84±0.71	33.00±1.15	63.49±3.10	75.95±0.99	30.40±2.74	66.44±0.69	74.47±0.62	70.25±1.29	88.41±1.01	95.95±0.77	89.62±1.81
GAAP	65.67±3.14	77.51±3.63	33.73±1.30	62.79±2.28	70.93±2.84	27.57±2.51	65.96±0.70	71.97±0.66	70.08±0.70	88.14±0.63	93.61±0.92	88.29±0.76
LLM	60.79±0.71	71.18±1.25	30.45±0.66	59.90±0.71	71.78±1.25	27.03±2.66	63.87±2.98	67.97±2.02	66.57±1.95	89.19±0.41	96.40±0.24	91.32±0.54
TAPE	64.33±1.81	74.00±1.41	37.89±2.03	63.25±1.61	74.43±1.81	31.84±1.64	66.76±0.83	70.66±1.10	68.78±1.70	90.12±1.29	96.53±1.86	92.10±1.37
FLAG	64.79±0.71	73.98±1.25	38.45±0.66	62.88±2.64	73.11±1.11	30.22±2.37	67.82±1.28	73.60±1.36	71.24±0.36	90.24±1.41	95.08±1.07	92.37±1.13
GraphGPT	60.96±0.99	70.39±1.19	30.66±1.48	59.13±2.06	70.76±2.09	27.82±1.90	64.12±1.74	67.38±1.40	67.85±2.50	87.54±2.42	91.62±1.85	87.59±2.74
HiGPT	62.49±0.58	71.80±0.78	31.59±0.24	60.99±1.85	71.49±2.30	29.69±2.37	66.70±0.67	69.80±0.24	67.62±0.16	89.61±2.08	96.15±2.37	90.15±1.70
InstructGLM	66.43±0.14	75.73±0.57	38.36±0.24	62.84±0.21	73.47±0.43	31.29±0.47	67.28±1.19	73.82±1.58	70.28±2.23	89.79±1.57	95.28±1.31	92.20±2.16
DGP	*69.07±0.23	*84.28±0.11	*48.87±0.82	*66.91±0.13	*77.32±0.11	*34.63±0.24	*75.01±0.11	*82.74±0.28	*82.35±0.20	*93.73±0.26	*98.04±0.06	*95.45±0.07

Table 2: Comparison of fraud detection performance (mean±std %) across datasets. The best and second-best results are marked in bold and underlined, respectively. * indicates the improvement over the second-best method is significant with $p < 0.05$.

forms on fraud detection tasks over real-world heterogeneous graphs compared to GNN and LLM baselines; (ii) *component analysis* — how major components affect the overall performance; and (iii) *parameter analysis* — how sensitive DGP is to hyperparameters and the design of summarization prompts.

Experimental Setup

Datasets We conduct experiments on four graph datasets, including two public benchmarks: YelpReviews (Rayana and Akoglu 2015) is a spam detection dataset in which each node represents a review labeled as spam or non-spam. We use its YelpChi subset and the associated raw texts for model evaluation. AmazonVideo (McAuley and Leskovec 2013) is a product review dataset for unhelpful review detection. We also perform evaluation on two proprietary industry datasets: LifeService and E-Commerce, which are real-world graphs sampled from our industry partner, ByteDance.

Table 1 presents the key properties of the datasets. All datasets are characterized by mixed textual and numerical features, multi-type edges, and imbalanced fraud labels. Given the high cost of manual annotation in industry settings, we construct training sets with limited labeled samples, simulating realistic constraints where high-quality fraud labels are costly and difficult to obtain. We also note that the sum of the dataset split sizes, including the training, validation, and test sets, can be smaller than the total number of nodes. This aligns with real-world scenarios in which the majority of nodes are unlabeled, leaving them outside the regular data splits.

Baselines We benchmark against a wide range of competitive models: (i) *GNNs*, including GraphSAGE (Hamilton, Ying, and Leskovec 2017), HGT (Hu et al. 2020), ConsisGAD (Chen et al. 2024), PMP (Zhuo et al. 2024), and GAAP (Duan et al. 2025); (ii) *Graph-agnostic models*, including MLP (Rosenblatt 1958) and a Qwen3-8B LLM (Team 2025) finetuned on target nodes alone; (iii) *LLM-enhanced GNNs*, represented by TAPE (He et al. 2024) and FLAG (Yang et al. 2025); and (iv) *graph-enhanced LLMs*, including GraphGPT (Tang et al. 2024a), HiGPT (Tang et al. 2024b), and InstructGLM (Ye et al. 2024). All baselines are implemented using official code.

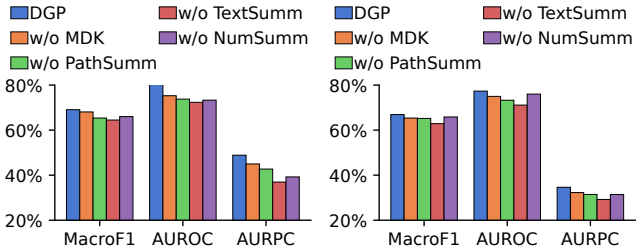
Parameter Settings For all evaluated models, we tune hyperparameters using grid search on the validation set.

Implementation Details. We conduct experiments using a Linux system with 64 Intel(R) Xeon(R) Gold 6346 CPUs, 1TB of RAM, and four NVIDIA A100 GPUs (80GB). For all LLM-tuning methods, we use the Qwen3-8B LLM backbone (Team 2025) for fair comparison. We apply LoRA (Hu et al. 2022) to all attention layers and use AdamW (Loshchilov and Hutter 2019) optimizer for fine-tuning. DGP additionally uses a frozen Qwen3-8B for summary generation. We adopt classification metrics, including Macro-F1, AUROC, and AUPRC, and report the mean and standard deviation over five random seeds. The model is implemented via PyTorch (Paszke et al. 2019), Transformers (Wolf et al. 2019), and DGL (Wang et al. 2019).

Performance Evaluation

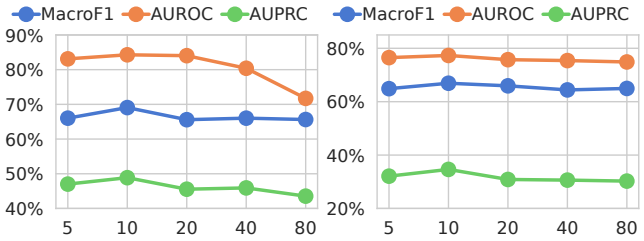
As shown in Table 2, DGP achieves consistent and substantial improvements over baselines across datasets and evaluation metrics. We make the following observations:

- GNNs outperform MLPs in most cases, demonstrating the importance of leveraging graph structural information for fraud detection in heterogeneous graphs. Advanced GNNs such as ConsisGAD, which incorporate more sophisticated structural modeling, achieve better performance by capturing complex graph semantics.
- Although the datasets contain rich textual information, recent standalone LLMs (without graph context) generally underperform MLPs and thus fail to effectively leverage the textual information for fraud detection.
- Methods like TAPE and InstructGLM combine graphs with LLMs, outperforming standalone LLMs by leveraging relational information. However, their performance remains constrained by challenges such as complex relations and neighborhoods in fraud detection scenarios, which can introduce noise and reduce effectiveness.



(a) YelpReviews (b) AmazonVideo

Figure 4: Ablation study on DGP components. "Path" denotes *Metapath*, "Num" denotes *Numerical*, and "Summ" denotes *Summarization*.



(a) YelpReviews (b) AmazonVideo

Figure 5: Impact of summarization budget (tokens).

Dataset	Task-Aware	Macro F1	AUROC	AUPRC
YelpReviews	✗	69.07±0.23	84.28±0.11	48.87±0.82
	✓	58.65±0.05	70.65±1.05	29.02±0.36
AmazonVideo	✗	66.91±0.13	77.32±0.11	34.63±0.24
	✓	65.55±0.21	73.73±0.17	31.82±0.27

Table 3: Impact of node-level summarization prompts.

- Compared to LLMs with encoding-based prompting, DGP avoids early vectorization and preserves more structural and textual information throughout the detection process. Compared to LLMs with text-only prompting, DGP uses dual-granularity summarization to reduce neighbor domination and information overload. These design choices enable DGP to achieve the best overall results. Notably, DGP surpasses the best GNN baselines, suggesting that LLMs, when properly enhanced with graph context and summarization, hold significant potential for graph-based fraud detection.

Detailed Analysis

Ablation Study To evaluate the effectiveness of each component in DGP, we conduct an ablation study on the YelpReviews and AmazonVideo datasets. As shown in Figure 4, we remove individual modules to obtain DGP variants and observe the resulting performance changes.

- Removing major components, including textual summarization (w/o TextSumm) or numerical summarization (w/o NumSumm), results in a clear performance drop. This demonstrates that these components are essential for capturing semantic and statistical signals in heterogeneous fraud-detection graphs. Specifically, textual summarization exhibits higher importance than numerical summarization, indicating that text in these datasets is particularly informative for fraud detection.
- We also ablate subsidiary components within textual summarization, including Markov Diffusion Kernel-based metapath trimming (w/o MDK) and metapath summarization (w/o PathSumm). Removing either component leads to a performance decline, indicating their positive contributions to generating informative representations of neighbor nodes.

Impact of Summarization Length We further examine the effect of varying the summarization budget B. For simplicity, we assume a unified budget, i.e., $B = B_{\text{node}} = B_{\text{meta}}$. Figure 5 presents fraud detection metrics across a range of budgets $B \in \{5, 10, 20, 40, 80\}$ for the YelpReviews and AmazonVideo datasets. We observe that very short summaries (5 tokens) provide insufficient context and thus degrade performance, while longer summaries (e.g., 80 tokens) may lead to token dilution, also reducing performance.

Notably, the best performance is generally achieved with a relatively small summarization budget (10 tokens). This suggests that highly coarse-grained summarizations of neighbor information are sufficient to enhance fraud detection on graphs. Importantly, this indicates that DGP remains effective in complex real-world graphs, as it does not rely on fine-grained or verbose descriptions of each neighbor.

Impact of Task-Aware Summarization We analyze whether task-aware summarization prompts help DGP's classification performance. Table 3 reports results for both task-agnostic and task-aware neighbor summarization strategies. In the task-agnostic setting, we use a generic instruction: `Summarize the text within 10 tokens`. In contrast, the task-aware setting introduces domain-specific cues, e.g., `Summarize the text within 10 tokens, focusing on signals indicative of fraudulent behavior`.

The results suggest that task-aware summarization degrades DGP's performance. The underlying reason could be reduced generality, where overly specific prompts constrain the LLM's ability to capture subtle fraud signals. In contrast, task-agnostic prompts allow for general cue discovery, potentially supporting more robust classification.

Conclusion

We introduced DGP, a framework for fraud detection on heterogeneous graphs that combines semantic-aware summarization, diffusion-based metapath trimming, and type-specific feature aggregation. By condensing relevant multi-hop textual contexts and precisely aggregating structured attributes, DGP enables effective fraud prediction using large language models. Extensive experiments demonstrate its superior performance and robustness across diverse benchmarks. In future work, we will explore dynamic graphs in which fraud patterns evolve over time.

Acknowledgments

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, and Ministry of Education, Singapore, under the Academic Research Fund Tier 2 (FY2025) (Grant MOE-T2EP20124-0009). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

References

Akoglu, L.; Tong, H.; and Koutra, D. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3): 626–688.

Barbero, F.; Banino, A.; Kapturowski, S.; Kumaran, D.; Madeira Araújo, J.; Vitvitskyi, O.; Pascanu, R.; and Veličković, P. 2024. Transformers need glasses! information over-squashing in language tasks. *Advances in Neural Information Processing Systems*, 37: 98111–98142.

Chen, N.; Liu, Z.; Hooi, B.; He, B.; Fathony, R.; Hu, J.; and Chen, J. 2024. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *The twelfth international conference on learning representations*.

Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; and Yu, P. S. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 315–324.

Duan, M.; He, D.; Zheng, T.; Jia, L.; Song, M.; Wang, X.; and Feng, Z. 2025. Global Attribute-Association Pattern Aggregation for Graph Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11616–11624.

Duan, M.; Zheng, T.; Gao, Y.; Wang, G.; Feng, Z.; and Wang, X. 2024. Dga-gnn: Dynamic grouping aggregation gnn for fraud detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 11820–11828.

Fatemi, B.; Halcrow, J.; and Perozzi, B. 2023. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*.

Fouss, F.; Francoisse, K.; Yen, L.; Pirotte, A.; and Saerens, M. 2012. An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. *Neural networks*, 31: 53–72.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, P.; Gao, J.; and Chen, W. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

He, X.; Bresson, X.; Laurent, T.; Perold, A.; LeCun, Y.; and Hooi, B. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. In *The Twelfth International Conference on Learning Representations*.

Hooi, B.; Shin, K.; Song, H. A.; Beutel, A.; Shah, N.; and Faloutsos, C. 2017. Graph-based fraud detection in the face of camouflage. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(4): 1–26.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Hu, J.; Hooi, B.; and He, B. 2024. Efficient heterogeneous graph learning via random projection. *IEEE Transactions on Knowledge and Data Engineering*.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, 2704–2710.

Huang, T.; and Wang, Y. 2025. Can LLMs Find Fraudsters? Multi-level LLM Enhanced Graph Fraud Detection. *arXiv preprint arXiv:2507.11997*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, K.; Yang, T.; Zhou, M.; Meng, J.; Wang, S.; Wu, Y.; Tan, B.; Song, H.; Pan, L.; Yu, F.; et al. 2024a. Sefraud: Graph-based self-explainable fraud detection via interpretative mask learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5329–5338.

Li, P.; Yu, H.; and Luo, X. 2025. Context-aware Graph Neural Network for Graph-based Fraud Detection with Extremely Limited Labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12112–12120.

Li, P.; Yu, H.; Luo, X.; and Wu, J. 2023a. LGM-GNN: A local and global aware memory-based graph neural network for fraud detection. *IEEE Transactions on Big Data*, 9(4): 1116–1127.

Li, S.; Yang, J.; Liang, G.; Li, T.; and Zhao, K. 2022. SybilFlyover: Heterogeneous graph-based fake account detection model on social networks. *Knowledge-Based Systems*, 258: 110038.

Li, T.; Zhang, G.; Do, Q. D.; Yue, X.; and Chen, W. 2024b. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Li, Y.; Li, Z.; Wang, P.; Li, J.; Sun, X.; Cheng, H.; and Yu, J. X. 2023b. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024a. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.

Liu, S.; Yao, D.; Fang, L.; Li, Z.; Li, W.; Feng, K.; Ji, X.; and Bi, J. 2024b. Anomalyllm: Few-shot anomaly edge detection for dynamic graphs using large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, 785–790. IEEE.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

McAuley, J. J.; and Leskovec, J. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, 897–908.

Niu, C.; Qiao, H.; Chen, C.; Chen, L.; and Pang, G. 2025. Zero-shot Generalist Graph Anomaly Detection with Unified Neighborhood Prompts. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, 3226–3234. ijcai.org.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qiao, H.; Niu, C.; Chen, L.; and Pang, G. 2025. AnomalyGFM: Graph foundation model for zero/few-shot anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2326–2337.

Rayana, S.; and Akoglu, L. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, 985–994.

Rosenblatt, F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6): 386.

Sharma, V.; Kumar, R.; Cheng, W.-H.; Atiquzzaman, M.; Srinivasan, K.; and Zomaya, A. Y. 2018. NHAD: Neuro-fuzzy based horizontal anomaly detection in online social networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(11): 2171–2184.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11): 992–1003.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024a. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Xia, L.; Yin, D.; and Huang, C. 2024b. Higpt: Heterogeneous graph language model. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2842–2853.

Team, Q. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vasylenko, P.; Treviso, M.; and Martins, A. F. 2025. Long-Context Generalization with Sparse Attention. *arXiv preprint arXiv:2506.16640*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, H.; Feng, S.; He, T.; Tan, Z.; Han, X.; and Tsvetkov, Y. 2023. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36: 30840–30861.

Wang, M.; Zheng, D.; Ye, Z.; Gan, Q.; Li, M.; Song, X.; Zhou, J.; Ma, C.; Yu, L.; Gai, Y.; et al. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xu, F.; Wang, N.; Wu, H.; Wen, X.; Zhao, X.; and Wan, H. 2024. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 9214–9222.

Yang, C.; Liu, H.; Wang, D.; Zhang, Z.; Yang, C.; and Shi, C. 2025. FLAG: Fraud Detection with LLM-enhanced Graph Neural Network. In *Proceedings of the 31st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '25)*.

Ye, R.; Zhang, C.; Wang, R.; Xu, S.; and Zhang, Y. 2024. Language is All a Graph Needs. *EACL*.

Yu, H.; Liu, Z.; and Luo, X. 2024. Barely supervised learning for graph-based fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16548–16557.

Zhu, H.; and Koniusz, P. 2021. Simple spectral graph convolution. In *International conference on learning representations*.

Zhu, X.; Xue, H.; Zhao, Z.; Xu, W.; Huang, J.; Guo, M.; Wang, Q.; Zhou, K.; and Zhang, Y. 2025. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*.

Zhuo, W.; Liu, Z.; Hooi, B.; He, B.; Tan, G.; Fathony, R.; and Chen, J. 2024. Partitioning message passing for graph fraud detection. *arXiv preprint arXiv:2412.00020*.