Bring Complex Geometric Information to LLMs: A Positional Survey of Graph Parametric Representation

Dongqi Fu[*], **Liri Fang**[*], **Zihao Li**[*], **Zhe Xu**[*], **Hanghang Tong, Vetle I. Torvik, Jingrui He**
University of Illinois Urbana-Champaign
{dongqif2, lirif2, zihaoli5, zhexu3, htong, vtorvik, jingrui}@illinois.edu

Abstract

Graphs, as a relational data structure, have been widely used in various application scenarios, such as molecule design and recommender systems. Recently, large language models (LLMs) are reorganizing in the AI community due to their strong reasoning and inference capabilities. Enabling LLMs to effectively process graph-structured data holds significant potential. Applications include: (1) distilling external knowledge bases to mitigate hallucination and overcome the context window limitation in retrieval-augmented generation; and (2) directly addressing graph-centric tasks such as protein design and drug discovery. However, feeding raw graph data into LLMs is impractical. Graphs often have complex topologies, large scale, and lack efficient semantic representations, all of which hinder their direct integration with LLMs. This raises a key question: can graph representations be expressed in natural language while still encoding rich structural and geometric information suitable for LLM input? One promising direction is the use of **graph parametric representation** or **graph law**. These approaches predefine a set of parameters (e.g., degree, diameter, temporal dynamics) and establish their values and relationships by analyzing distributions across real-world graphs. Such parametric representations may offer a natural bridge for LLMs to understand complex graph structures and perform corresponding inferences. Therefore, in this survey, we first review four categorical of current efforts of incorporating graph data into LLMs, i.e., topological query, semantic query, GNN embedding, and GNN prediction, highlighting their limitations. Then, we introduce graph parametric representation from multiple perspectives, including macroscopic vs. microscopic views, low-order vs. high-order structures, and static vs. temporal graphs. Finally, we conclude the paper with future research directions. Resource: https://github.com/DongqiFu/Awesome-Graphs-into-LLMs

1 Introduction

Graphs serve as a fundamental relational data structure and are extensively utilized in a wide range of application scenarios, including molecule design, social network analysis, and recommender systems [75]. Recent efforts combine geometric message passing with language models [43, 67], e.g., 3D material generation [62], drug design [16], and symbolic–geometric reasoners [38]. Their ability to represent complex interconnections among entities makes them indispensable in modeling real-world relationships. However, despite their widespread use, integrating graph-based data input with large language models (LLMs) remains a challenging problem.

Recently, LLMs have revolutionized the AI community with their remarkable reasoning and inference capabilities [18, 54]. These models have demonstrated significant potential in various tasks, including natural language understanding, machine translation, and knowledge extraction. Given the growing importance of LLMs, enabling them to comprehend and process graph-based relational data could

[*] Equal Contribution

Fu et al., Bring Complex Geometric Information to LLMs: A Positional Survey of Graph Parametric Representation. *Proceedings of the Fourth Learning on Graphs Conference (LoG 2025)*, PMLR 269, Hybrid Event, December 10–12, 2025.

open new frontiers in artificial intelligence research and applications. This integration holds immense potential for enhancing LLMs in multiple ways, including but not limited to:

- **Knowledge Distillation for LLMs**: Graph-based external knowledge bases can provide crucial insights, mitigating issues such as hallucinations in LLM-generated responses and overcoming the limitations imposed by fixed context windows. By incorporating structured graph data, LLMs can improve retrieval-augmented generation (RAG) techniques and enhance inference accuracy [13, 22].
- **Direct Graph-Based Problem Solving**: Many research domains, such as protein design and drug discovery, inherently rely on graph-based data representations [39, 57]. Equipping LLMs with the capability to understand and manipulate graph structures could significantly advance research in these fields by enabling direct problem-solving approaches.

Despite the clear advantages of incorporating graph data into LLMs, several challenges hinder this integration. The primary obstacles include (1) the complexity of graph topologies, (2) the size of graph datasets, and (3) the absence of effective semantic representations of graphs that LLMs can process efficiently. Unlike textual data, which LLMs are inherently designed to understand, graphs lack a straightforward natural language representation. This leads to a fundamental research question: Is there a form of graph representation that is both interpretable in natural language for LLMs and informative enough to serve as a viable input format?

A promising solution lies in the concept of **graph parametric representation** or **graph laws** [17], which refers to statistical principles that define relationships between key structural parameters of graphs, such as degree, clustering coefficients, diameter, and time, i.e., a graph is expected to be represented by a few parameters to reflect its properties well. The formal mathematical relations and concrete values of the parameters are estimated by analyzing real-world and large-scale graph data distributions [33, 34]. By encoding graph properties through the pre-defined set of parameters, graph laws offer a way to translate complex graph topologies into a form that LLMs can potentially comprehend, e.g., the relationship between the possibility of a newly arrived node connecting to an old node (parameter #1) and the degree of that old node (parameter #2) is determined by maximum likelihood estimation (MLE) based on the observed real-world graph data.

In the rest of the paper, in Section 2, we first summarize the current efforts in incorporating graph data into LLMs, categorizing them into fourfold, and discuss the corresponding limitations to recall the role of parametric representation in the era of LLMs. Before we introduce the specific graph laws and representation parameters in Section 4 and Section 5, we systematically illustrating the pathway of bringing graph parametric representations into LLMs in Section 3 We introduce related work in Section 6, highlight a few future directions on graph parametric representation study in Section 7, conclude the paper in Section 8, and leave some newly discovered graph parameters in Appendix A.

2 Current Efforts for Incorporating Graphs into LLMs

In this section, we first disentangle four theoretical approaches for incorporating graph data as input to LLMs. We then comprehensively present various methods within this category in Table 2, with the symbols summarized in Table 1.

Table 1: Notation

Symbol	**Description**
$G = (V, E)$	Graph with node set V and edge set E
v_i	The i-th node
T_i	Text features (e.g., words, sentences, paragraphs, etc.) of the i-th node
$\mathcal{N}(i)$	The set of 1-hop neighbors of node i
$\mathbf{A} \in \{0, 1\}^{\lvert V\rvert \times \lvert V\rvert}$	Adjacency matrix of graph G

2.1 Topological Query

A **topological query** (i.e., `Strategy` ①) represents a direct and intuitive approach to converting graph data into textual input for LLMs. For example, for node-level tasks such as node classification, the textual features of the target (center) node along with its 1-hop and 2-hop neighbors [41, 64] are concatenated to form the input sequence; Eq. (1) shows the concatenated textual features from node i's 1-hop neighbors,

$$T_{\mathcal{N}(i)} = \texttt{Concat}(\{T_j : j \in \mathcal{N}(i)\}) \tag{1}$$

where $\texttt{Concat}(\cdot)$ is the function for text concatenation.

We refer to this as a topological query because the subgraph selection relies solely on structural properties of the graph. Notable techniques under this category include Personalized PageRank [20], which ranks nodes based on their topological relevance to the query node by iteratively calling Eq. (2):

$$\mathbf{r}_i \leftarrow (1-\alpha)\tilde{\mathbf{A}}\mathbf{r}_i + \alpha\mathbf{q}_i \tag{2}$$

where $\tilde{\mathbf{A}}$ is the normalized adjacency matrix (often row-stochastic); α is the teleport probability (restart probability); $\mathbf{q}_i \in \{0,1\}^{|V|}$ is a vector with a 1 at position i and 0 elsewhere. Top-K nodes' textual features are concatenated as

$$T_{\text{top-}K(i)} \;=\; \texttt{Concat}(\{T_j : j \in \text{Top-}K(\mathbf{r}_i)\})$$

For graph-level tasks, the most straightforward approach is to present the graph as a node list and an edge list, allowing this textual sequence to be used as input for LLMs. More formal option to define or represent graph-structured data includes graph description language such as Graph Modeling Language (GML) [24] and Graph Markup Language (GraphML) [5].

2.2 Semantic Query

A **semantic query** (i.e., Strategy ②) aims to retrieve subgraphs based on semantic relevance rather than topological proximity. This approach is inspired by retrieval-augmented generation (RAG) methods [30, 35], which enhance model performance by integrating retrieved information. In this context, for a given center node or subgraph, semantically related nodes are identified [61, 74], and their textual features are concatenated to construct the LLM input.

Concretely, given a target node v_i and its textual feature T_i, the most semantically relevant node is retrieved by

$$\arg\max_{j\neq i} \; \langle \texttt{Encoder}(T_i), \texttt{Encoder}(T_j) \rangle \tag{3}$$

where Encoder is a text encoder such as sentence-BERT [48] and $\langle\cdot,\cdot\rangle$ denotes the inner product.

2.3 Graph Neural Network as Encoder

Applying graph neural networks (GNNs) to encode graph structures is a natural and widely adopted strategy. For example, the node i's embedding from the k-th layer of a message-passing neural network can be presented as

$$\mathbf{h}_i^{(k)} = \sigma\left(\mathbf{W}^{(k)} \cdot \texttt{AGG}\Big(\{\, \mathbf{h}_j^{(k-1)} : j \in \mathcal{N}(i) \,\}\Big)\right) \tag{4}$$

$$\mathbf{h}_i^{(0)} = \texttt{Encoder}(T_i) \tag{5}$$

where AGG denotes the aggregation operator such as sum; $\mathbf{W}^{(k)}$ is the weight matrix from the k-th layer; σ is the activation function.

As **GNNs produce latent representations** $\{\mathbf{h}_i\}_{i\in V}$ (i.e., Strategy ③) that are not easily translatable into text, these embeddings are often injected into LLMs [55, 76] via latent-layer integration techniques such as soft prompting [7, 64]. Alternatively, some methods utilize interpretable **GNN predictions in textual form** (i.e., Strategy ④) such as Eq. (6) as a direct input to LLMs [61], enabling a different mode of interaction between the two models.

$$\ell_i \;=\; \texttt{LabelMap}\left(\arg\max_{c\in\{1,\ldots,C\}} \mathbf{h}_i(c)\right) \tag{6}$$

which reads the largest logit from the graph neural network's output $\mathbf{h}_i$ and translates it into the textual label through the LabelMap function; C is the total number of labels.

Table 2 presents a comparative overview of existing models, emphasizing how each method incorporates graph data into LLMs. It is important to note that this survey focuses on frameworks where the **LLM serves as the primary task solver**; approaches that employ LLMs solely to assist or enhance GNNs fall outside the scope of this work.

Table 2: Comparison of Language Models in their Use of Graph Data as Input. Strategies 1 to 4 denotes ① topological query, ② semantic query, ③ GNN embedding, ④ GNN prediction

Method	Ref	Year	Backbone LLM	Fine Tuning	Strategies				Task Level
					①	②	③	④	
NLGraph	[56]	2023	GPT	No	✓				Node, Link, Graph
GPT4Graph	[19]	2023	GPT	No	✓				Node, Link, Graph
LLM4GT	[50]	2023	GPT	No	✓				Node, Link
GraphText	[73]	2023	GPT	No	✓				Node
DGTL	[47]	2023	Llama	Yes	✓		✓		Node
InstructGLM	[64]	2024	T5, Llama	Yes	✓		✓		Node
LlaGA	[7]	2024	Llama	Yes	✓		✓		Node, Link
AuGLM	[61]	2024	T5	Yes	✓	✓		✓	Node
GraphPrompter	[41]	2024	Llama	Yes			✓		Node, Link
G-Recall	[59]	2024	GPT, Gemini	No	✓				Subgraph
TLG	[15]	2024	PaLM	No	✓				Node, Link, Subgraph
LLM4DyG	[71]	2024	GPT, Llama	No	✓				Temporal
SNS	[37]	2024	GPT	No		✓			Node
MuseGraph	[52]	2024	BART, T5, Llama	Yes	✓				Node, Graph
OFA	[40]	2024	Llama	No	✓				Node, Link, Graph
GraphGPT	[53]	2024	Llama	Yes	✓		✓		Node, Link
Graph-CoT	[29]	2024	GPT	No	✓				Node
TEA-GLM	[55]	2024	Vicuna	Yes			✓		Node, Link
GPEFT	[76]	2024	Llama	Yes			✓		Link
PromptGFM	[77]	2025	T5, Llama	Yes	✓				Node, Link
GraphICL	[51]	2025	Llama, GPT	No	✓	✓			Node, Link
UniGraph	[23]	2025	Llama	Yes	✓		✓		Node, Link, Graph
SKETCH	[74]	2025	Nomic, Llama	Yes	✓	✓			Node
TGTalker	[27]	2025	Qwen, Mistral, Llama	No	✓				Temporal
FewshotRAG	[36]	2025	Llama, Qwen, etc.	No		✓			Node

2.4 Limitations by Complex Geometric Information

Recent studies suggest that current methods of incorporating graph data into LLMs are insufficient for enabling deep graph understanding. Several works [15, 19, 56] indicate that LLMs exhibit only limited graph reasoning capabilities, performing weakly on fundamental tasks such as graph size estimation, degree computation, and edge existence detection. Furthermore, advanced prompting strategies, including chain-of-thought prompting, tend to be less effective when applied to more complex graph problems. Similar limitations are also observed in dynamic graph settings [71]. Additional evidence [26] suggests that LLMs may interpret graph-structured input merely as sequential text, lacking an understanding of the underlying structure. The Graph Recall Test, a simple yet revealing benchmark, further demonstrates that most LLMs fail to retain and reason over graph information reliably [59]. Moreover, according to [61], LLM-based node classifiers still significantly underperform compared to specialized GNN-based classifiers, and an intuitive guess is that this is due to the GNN's ability in the Weisfeiler-Lehman graph isomorphism test [60]. Therefore, we want to ask, *Is there a carrier that can bring complex geometric information to LLMs?*

3 Bring Graph Parametric Representations to LLMs

Before Sections 4 and 5 outlining the specific graph law and representation parameters, an essential step is to translate these theoretical regularities into a form that large language models (LLMs) can internalize and reason with. Bridging the two domains requires a representation layer that is both parametrically compact and semantically aligned with the textual interface of LLMs. We systematically illustrate this pathway from graph-theoretic priors to language-based reasoning, with a conceptual example as shown in Figure 1.

```
GraphSummary:
- name: "example_graph"
- nodes: 102,345
- edges: 1,234,567
- densification_exponent_alpha: 1.12
- effective_diameter_90pct: 6.3
- avg_clustering_coef: 0.21
- degree_distribution: "heavy-tailed (power-law, gamma =
   2.8)"
- motif_counts: {"triangle": 12345, "4-cycle": 2345}
- spectral_gap (second smallest eigenvalue): 0.015
- temporal_window: "2016-01-01 -- 2020-12-31"

Task: "Using the GraphSummary above, estimate
[node classification / link prediction / ...]
or answer Q: ...
Provide reasoning and cite which parameter(s) you used."
```

Figure 1: Example prompt for LLM-based graph reasoning. Graph parameters are serialized as a structured text block, providing interpretable context for downstream tasks.

From Empirical Laws to Parametric Descriptors (`Step` ①). Graph laws such as densification or degree distributions encode families of structural invariants. Each law can be summarized by a small set of numeric parameters—e.g., the densification exponent, effective diameter, or average clustering—that capture global topology while remaining independent of specific node identities. We can refer to this vector of key statistics as a graph parametric summary, which is expect to serve as a low-entropy bottleneck that condenses complex geometry into interpretable quantitative cues.

Language Grounding via Symbolic Templates (`Step` ②). To make these summaries accessible to LLMs, the parameters can be serialized into symbolic or natural-language templates. For example: *Graph A has n = 300 nodes, average degree = 10, clustering coefficient 0.18, and diameter = 7.* Such textualized forms preserve the semantics of the underlying law while aligning with the token-based processing of LLMs. Compared with raw adjacency lists or embeddings, they strike a balance between interpretability and information sufficiency.

Contextual Injection Strategies (`Step` ③). Parametric summaries can be injected into LLMs by:

- *Prompt-Level Conditioning*, where summaries are prepended as context before reasoning questions;
- *Retrieval-Augmented Prompting*, in which graph laws most relevant to the query are dynamically retrieved and inserted;
- *Adapter-based Fine-Tuning*, where the parametric vector is converted into soft tokens or key-value biases for the model's attention layers.

Reasoning Alignment and Interpretability (`Step` ④). Because each parameter has an interpretable geometric meaning, the resulting reasoning chains become explainable: LLMs can ground relational claims ("Graph A is denser but has a smaller diameter than Graph B") in quantitative laws. This alignment bridges continuous geometry and discrete language, offering a principled route to geometric interpretability in LLM reasoning.

4 Macroscopic Graph Parametric Representation

In this section, we introduce the graph laws from the macroscope and microscope. In detail, we will introduce the intuition behind researchers proposing or using graph statistical properties as parameters and how they fit the values of these parameters against real-world observations.

Table 3: A summary of parameteric representations of graphs. Some laws have multiple aspects and are indexed by numbers in parentheses.

Input	Law	Parameter	Scope	Order	Temporality	Description
Graphs	Densification Law [34]	Density degree α	Macro	Low	Dynamic	$e(t) \propto n(t)^\alpha, \alpha \in [1, 2]$, $e(t)$ is # edges at t
	Shrinking Law [34]	Effective diameter d	Macro	Low	Dynamic	$d_{t+1} < d_t$, d decreases as network grows
	Motif Differing Law(1) [45]	Numbers of similar motifs n	Macro	High	Dynamic	$n_1 \neq n_2$ for different domains
	Motif Differing Law(2) [45]	Motif occurring timestamp t	Macro	High	Dynamic	$t_1 \neq t_2$ for different motifs
	Egonet Differing Law [4]	Features of Egonets X	Macro	High	Static	$X_1 \neq X_2$ for different domains
	Simplicial Closure Law [4]	Simplicial closure probability p	Macro	High	Static	p increases with additional edges or tie strength
	Spectral Power Law(1) [14]	Degree, SVD, eigen distributions	Macro	High	Static	These distributions usually follow power-law
	Spectral Power Law(2) [14]	Degree, SVD, eigen distributions	Macro	High	Static	If one follow power-law, usually others follow
	Edge Attachment Law(1) [33]	Node degree d, edge create $p_e(d)$	Micro	Low	Dynamic	$p_e(d) \propto d$ for node with degree d
	Edge Attachment Law(2) [33]	Node age $a(u)$, edge create $p_e(d)$	Micro	Low	Dynamic	$p_e(d)$ seems to be non-decreasing with $a(u)$
	Triangle Closure Law(1) [25]	Triangular connections e_1, e_2, e_3	Micro	Low	Dynamic	Strong $e_3 \Rightarrow$ unlikely e_1/e_2 will be weakened
	Triangle Closure Law(2) [25]	Triangular connections e_1, e_2, e_3	Micro	Low	Dynamic	Strong $e_1/e_2 \Rightarrow$ unlikely they will be weakened
	Local Closure Law [66]	Local closure coefficient $H(u)$	Micro	Low	Static	Please refer to Section A for details
	Spectral Density Law [10]	Density of states $\mu(\lambda)$	Macro	High	Static	Please refer to Section A for details
	Motif Activity Law(1) [70]	Motif type	Micro	High	Dynamic	Motifs do not transit from one type to another
	Motif Activity Law(2) [70]	Motif re-appear rate	Micro	High	Dynamic	Motifs re-appear with configured rates
Hypergraphs	Degree Distribution Law [9]	Node degree, edge link probability	Macro	High	Dynamic	High-degree nodes are likely to form new links
	SVD Distribution Law [9]	Singular value distribution	Macro	High	Static	Singular value distribution usually heavy-tailed
	Diminishing Overlaps [31]	density of interactions $DoI(\mathcal{H}(t))$	Macro	High	Dynamic	Overall hyperedge overlaps decrease over time
	Densification Law [31]	Density degree α	Macro	High	Dynamic	$e(t) \propto n(t)^\alpha, \alpha \geq 1$, $e(t)$ is # hyperedges at t
	Shrinking Law [31]	Hypergraph effective diameter d	Macro	High	Dynamic	$d_{t+1} < d_t$, d decreases as network grows
	Edge Interacting Law [8]	Edge interacting rate	Micro	High	Dynamic	Temporally adjacent interactions highly similar
Heterographs	Densification Law [58]	Density degree α, # meta-path	Macro	Low	Dynamic	$e(t) \propto n(t)^\alpha, \alpha \geq 1$ for some meta-path
	Non-densification Law [58]	Density degree α, # meta-path	Macro	Low	Dynamic	Maybe, for some meta-path, $e(t) \not\propto n(t)^\alpha$

Several classical theories model the growth of graphs. For example, the Barabasi-Albert model [2, 3] assumes that graphs follow a uniform growth pattern in terms of the number of nodes. The Bass model [42] and the Susceptible-Infected model [1] follow the Sigmoid growth (more random graph models can be found in [12]). However, these pre-defined graph growths have been tested, and they could not handle the complex real-world network growth patterns very well [32, 69]. To this end, researchers begin to fit the graph growth on real-world networks directly to discover graph laws.

4.1 Low-Order Macroscopic Parametric Representation

Based on fitting nine real-world temporal graphs from four different domains, the authors in [34] found two temporal graph laws, called (1) *Densification Laws* and (2) *Shrinking Diameters*, respectively. First, the densification law states as follows.

$$e(t) \propto n(t)^\alpha \tag{7}$$

where $e(t)$ denotes the number of edges at time t, $n(t)$ denotes the number of nodes at time t, $\alpha \in [1, 2]$ is an exponent representing the density degree. The second law, shrinking diameters, states that *the effective diameter is decreasing as the network grows, in most cases*. Here, the diameter means the node-pair shortest distance, and the effective diameter of the graph means the minimum distance d such that approximately 90% of all connected pairs are reachable by a path of length at most d. Later, in [69], the densification law gets in-depth confirmed on four different real social networks, the research shows that the number of nodes and number of edges both grown exponentially with time, i.e., following the power-law distribution.

4.2 High-Order Macroscopic Parametric Representation

Above discoveries are based on the node-level connections (i.e., low-order connections). Several researchers start the investigation based on the group activities, for example, motifs [45], simplices [4], and hyperedges [9, 31]. Motif is defined as a subgraph induced by a sequence of selected temporal edges in [45], where the authors discovered that *different domain networks have significantly different numbers of similar motifs, and different motifs usually occur at different time*. Similar laws are also discovered in [4], where the authors study 19 graph data sets from domains such as biology, medicine, social networks, and the web to characterize how high-order structure emerges and differs across different domains. They discovered that the higher-order Egonet features can discriminate the domain of the graph, and the probability of simplicial closure events typically increases with the addition of edges or tie strength.

In hypergraphs, each hyperedge could connect an arbitrary number of nodes, rather than two [9], where the authors found that real-world static hypergraphs obey the following properties: (1) *Giant Connected Components*, that there is a connected component comprising a large proportion of nodes, and this proportion is significantly larger than that of the second-largest connected component. (2) *Heavy-Tailed Degree Distributions*, that high-degree nodes are more likely to form new links. (3) *Small Effective Diameters*, that most connected pairs can be reached by a small distance (4) *High Clustering Coefficients*, that the global average of local clustering coefficient is high. (5) *Skewed Singularvalue Distributions*, that the singular-value distribution is usually heavy-tailed. Later, the evolution of real-world hypergraphs is investigated in [31], and the following laws are discovered.

- *Diminishing Overlaps*: The overall overlaps of hyperedges decrease over time.
- *Densification*: The average degrees increase over time.
- *Shrinking Diameter*: The effective diameters decrease over time.

To be specific, given a hypergraph $G(t) = (V(t), E(t))$, the density of interactions is stated as

$$DoI(G(t)) = \frac{|\ \{\{e_i, e_j\} \mid e_i \cap e_j \neq \emptyset \text{ for } e_i, e_j \in E(t)\}\ |}{|\{\{e_i, e_j\}|e_i, e_j \in E(t)\}|} \tag{8}$$

and the densification is stated as

$$|E(t)| \propto |V(t)|^s \tag{9}$$

where $s > 1$ stands for the density term, which echos the law discovered in the low order [34] as expressed in Eq 7.

In heterogeneous information networks (where nodes and edges can have multiple types), the power law distribution is also discovered [58]. For example, for the triplet "author-paper-venue" (i.e., A-P-V), the number of authors is power-law distributed with respect to the number of A-P-V instances composed by an author.

5 Microscopic Graph Parametric Representation

In contrast to representing the distribution of the entire graph, many researchers try to model individual behavior and investigate how they interact with each other to see the evolution pattern microscopically.

5.1 Low-Order Microscopic Parametric Representation

In [33], the authors view temporal graphs in a three-fold process, i.e., node arrival (determining how many nodes will be added), edge initiation (how many edges will be added), and edge destination (where each edge will be added). They ignore the deletion of nodes and edges, and they assign variables (models) to parameterize this process.

- *Edge Attachment with Locality* (an inserted edge closing an open triangle): It is responsible for the edge destination.
- *Node Lifetime* and *Time Gap between Emitting Edges*: It is responsible for edge initiation.
- *Node Arrival Rate*: It is responsible for the node arrival.

To model individual behaviors, there are many candidate models to select from. For example, in edge attachment, the probability of a newcomer u connecting to a node v can be proportional to v's current degree, v's current age, or a combination of both. Based on fitting each model to the real-world observation under the supervision of the MLE principle, the authors empirically choose the *random-random* model for edge attachment with locality, i.e., first, let node u choose a neighbor

v uniformly and let v uniformly choose u's neighbor w to close a triangle. The node lifetime and time gap between emitting edges are defined as follows.

$$a(u) = t_{d(u)}(u) - t_1(u) \tag{10}$$

where $a(u)$ stands for the age of node u, $t_k(u)$ is the time when node u links its k^{th} edge, $d_t(u)$ denote the degree of node u at time t, and $d(u) = d_T(u)$. T is the final timestamp of the data.

$$\delta_u(d) = t_{d+1}(u) - t_d(u) \tag{11}$$

where $\delta_u(d)$ records the time gap between the current time and the time when that node emits its last edge. Finding the node arrival is a regression process in [33], for example, in Flickr graph $N(t) = exp(0.25t)$, and $N(t) = 3900t^2 + 76000t - 130000$ in LinkedIn graph.

In [46, 63], the selection of edge attachment has flourished, where the authors propose several variants of edge attachment models for preserving graph properties. Regarding the triangle closure phenomenon, several in-depth research follow-ups have been conducted. For example, in [25], researchers found that (1) *the stronger the third tie (the interaction frequency of the closed edge) is, the less likely the first two ties are weakened*; (2) *when the stronger the first two ties are, the more likely they are weakened*.

5.2 High-Order Microscopic Parametric Representation

Hypergraph ego-network [8] is a structure defined to model the high-order interactions involving an individual node. The star ego-network $T(u)$ is defined as follows.

$$T(u) = \{s : (u \in s)\}, \forall s \in S \tag{12}$$

where S is the set of all hyperedges (or simplices). Also, in [8], there are other hypergraph ego-networks, like radial ego-network $R(u)$ and contracted ego-network $C(u)$. The relationship between them is as follows.

$$T(u) \subseteq R(u) \subseteq C(u) \tag{13}$$

In [8], authors observe that contiguous hyperedges (simplices) in an ego-network tend to have relatively large interactions with each other, which suggests that *temporally adjacent high-order interactions have high similarity, i.e., the same nodes tend to appear in neighboring simplices*.

In [70], authors try to model the temporal graph growth in terms of motif evolution activities. In brief, this paper investigates how the number of motifs changes and what the exact motif types are in each time interval, and fits the arrival rate parameter of each type of motif against the entire observed temporal graph.

6 Related Work

To the best of our knowledge, there are only a few survey papers on graph laws, with none published after 2022, marking the beginning of the foundation model era. A 2006 survey [6] primarily focused on graph laws for mining patterns, discussing the Densification Law and Shrinking Law. In 2016, another survey [11] shifted its focus towards the generation of large graphs using various graph modeling methods, including the Erdős-Rényi model, Watts-Strogatz model, and Albert-Barabási model. More recently, in 2019, the authors in [12] offered a broader perspective on random graph modeling, covering generative, feature-driven, and domain-specific approaches. In contrast to these earlier surveys, which were published before the advent of graph neural networks and prior to the discovery of several significant graph laws [8–10, 31, 58, 66, 70], our work represents the first survey to explore the potential of graph laws in the context of foundation models. We emphasize how graph laws can address domain inconsistencies across different graph data types and contribute to multimodal representation learning. Additionally, this survey is the first to offer an overview of high-order graph laws and heterogeneous graph laws, marking a novel contribution to the literature.

7 Future Directions

Here, we outline several interesting research directions in graph parametric representation within modern graph research.

Graph Laws on Temporal Graphs. Discovering accurate temporal graph laws from real-world networks heavily relies on the number and size of networks (e.g., the number of nodes, edges, and time duration). However, some of the temporal graph law studies mentioned above typically consider the number of graphs ranging from 10 to 20 when discovering the evolution pattern. The existence of time-dependent structure and feature information increases the difficulty of collecting real-world temporal graph data. To obtain robust and accurate (temporal) graph laws, we may need a considerably large amount of (temporal) network data available. Fortunately, we have seen some pioneering work, such as TGB [28] and TUDataset [44].

Graph Laws on Heterogeneous Networks. Although many graph laws have been proposed and verified on homogeneous graphs, real-world networks are typically heterogeneous [49] and comprise a large number of interacting, multi-typed components. While the existing work [58] only studied 2 datasets to propose and verify the heterogeneous graph power law, the potential exists for a transition in graph laws from homogeneous networks to heterogeneous networks, suggesting the presence of additional parameters contributing to the comprehensive information within heterogeneous networks. For example, in an academic network, the paper citation subgraph and the author collaboration subgraph may have their own subgraph laws that affect the laws of other subgraphs. Furthermore, Knowledge graphs, as a special group of heterogeneous networks, have not yet attracted much attention from the research community to study their laws.

Transferability of Graph Laws. As we can see in the front part of the paper, many nascent graph laws are described verbally without the exact mathematical expression, which hinders the transfer from the graph law to the numerical constraints for the representation learning process. One latent reason for this phenomenon is that selecting appropriate models and parameters, and fitting the exact values of these parameters to large evolving graphs, is very computationally demanding.

Taxonomy of Graph Laws. After we discovered many graph laws, is there any taxonomy or hierarchy of those? For example, graph law A stands in the superclass of graph law B, and when we preserve graph law A during the representation, we actually have already preserved graph law B. For example, a hierarchy of different computer vision tasks has recently been discovered [68]. Corresponding research on graph law development appears to be a promising direction.

Domain-Specific Graph Laws. Since graphs serve as general data representations with extreme diversity, it is challenging to find universal graph laws that fit all graph domains because each domain may be internally different from another [72]. In fact, in many cases, we have prior knowledge about the domain of a graph, which can be a social network, a protein network, or a transportation network. Thus, it is possible to study domain-specific graph laws that work well on only a portion of graphs and then apply these laws specifically to those graphs.

LLMs as GNNs. In the background of large language models (LLMs) development, an interesting question attracts considerable research interest nowadays, i.e., **can LLMs replace GNNs as the backbone model for graphs?** To answer this question, many recent works show the great efforts [21, 26, 65], where the key point is how to represent the structural information as the input for LLMs.

For example, Instruct-GLM [65] follows the manner of instruction tuning and makes the template $\mathcal{T}$ of a 2-hop connection for a *central node* v as follows.

$$\mathcal{T}(v, \mathcal{A}) = \{v\} \text{ is connected with } \{|v_2|_{v_2 \in \mathcal{A}_2^v}\} \text{ within two hops.} \tag{14}$$

where $\mathcal{A}_k^v$ represents the list of node v's k-hop neighbors.

As discussed above, the topological information (e.g., 1-hop or 2-hop connections) can serve as external modality information to contribute to (e.g., through prompting) the reasoning ability of large language models (LLMs) [26] and achieve state-of-the-art on low-order tasks like node classification and link prediction.

8 Conclusion

Motivated by the need for LLMs to understand graphs, we first categorize the current efforts, and then review the concepts and development progress of graph parametric representations (i.e., graph laws) from different perspectives, including macro- and microscopes, low-order and high-order connections, and static and temporal graphs. Finally, we envision the latent challenges and opportunities of graph parametric representations in modern graph research with several interesting and possible future directions.

Acknowledgment

This work is supported by National Science Foundation under Award No. IIS-2416070 and IIS-2433308. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

[1] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992. 6

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999. 6

[3] Albert-Laszlo Barabâsi, Hawoong Jeong, Zoltan Néda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3-4):590–614, 2002. 6

[4] Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon M. Kleinberg. Simplicial closure and higher-order link prediction. *Proc. Natl. Acad. Sci. USA*, 115(48):E11221–E11230, 2018. 6, 7

[5] Ulrik Brandes, Markus Eiglsperger, Ivan Herman, Michael Himsolt, and M. Scott Marshall. Graphml progress report: Structural layer proposal. In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing (GD 2001)*, volume 2265 of *Lecture Notes in Computer Science*, pages 501–512. Springer, Berlin, Heidelberg, 2002. 3

[6] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006. 8

[7] Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 3, 4

[8] Cazamere Comrie and Jon Kleinberg. Hypergraph ego-networks and their temporal evolution. *arXiv preprint arXiv:2112.03498*, 2021. 6, 8

[9] Manh Tuan Do, Se-eun Yoon, Bryan Hooi, and Kijung Shin. Structural patterns and generative models of real-world hypergraphs. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD 2020*, pages 176–186. ACM, 2020. 6, 7

[10] Kun Dong, Austin R. Benson, and David Bindel. Network density of states. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *KDD 2019*, pages 1152–1161. ACM, 2019. 6, 8, 15

[11] Georgios Drakopoulosa, Stavros Kontopoulosb, Christos Makrisb, and Vasileios Megalooikonomoua. Large graph models: A review. *arXiv preprint arXiv:1601.06444*, 2016. 8

[12] Mikhail Drobyshevskiy and Denis Turdakov. Random graph modeling: A survey of the concepts. *ACM Comput. Surv.*, 52(6):131:1–131:36, 2020. 6, 8

[13] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. 2

[14] Nicole Eikmeier and David F. Gleich. Revisiting power-law distributions in spectra of real world networks. In *KDD 2017*, pages 817–826. ACM, 2017. 6, 14

[15] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 4

[16] Cong Fu, Xiner Li, Blake Olson, Heng Ji, and Shuiwang Ji. Fragment and geometry aware tokenization of molecules for structure-based drug design using language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 1

[17] Dongqi Fu and Jingrui He. Natural and artificial dynamics in graphs: Concept, progress, and future. *Frontiers Big Data*, 5, 2022. 2

[18] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1

[19] Jiayan Guo, Lun Du, and Hengyu Liu. Gpt4graph: Can large language models understand graph structured data ? an empirical evaluation and benchmarking. *CoRR*, abs/2305.15066, 2023. 4

[20] Taher H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002. 3

[21] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 9

[22] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*, 2024. 2

[23] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In Yizhou Sun, Flavio Chierichetti, Hady W. Lauw, Claudia Perlich, Wee Hyong Tok, and Andrew Tomkins, editors, *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pages 448–459. ACM, 2025. 4

[24] Michael Himsolt. Gml: A portable graph file format. Technical report, Universität Passau, 1997. Technical Report. 3

[25] Hong Huang, Yuxiao Dong, Jie Tang, Hongxia Yang, Nitesh V. Chawla, and Xiaoming Fu. Will triadic closure strengthen ties in social networks? *ACM Trans. Knowl. Discov. Data*, 12(3):30:1–30:25, 2018. 6, 8

[26] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. Can llms effectively leverage graph structural information: When and why. *CoRR*, abs/2309.16595, 2023. 4, 9

[27] Shenyang Huang, Ali Parviz, Emma Kondrup, Zachary Yang, Zifeng Ding, Michael M. Bronstein, Reihaneh Rabbany, and Guillaume Rabusseau. Are large language models good temporal graph learners? *CoRR*, abs/2506.05393, 2025. 4

[28] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael M. Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. *CoRR*, abs/2307.01026, 2023. 9

[29] Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 163–184. Association for Computational Linguistics, 2024. 4

[30] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020. 3

[31] Yunbum Kook, Jihoon Ko, and Kijung Shin. Evolution of real-world hypergraphs: Patterns and models without oracles. In Claudia Plant, Haixun Wang, Alfredo Cuzzocrea, Carlo Zaniolo, and Xindong Wu, editors, *ICDM 2020*, pages 272–281. IEEE, 2020. 6, 7, 8

[32] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue B. Moon. What is twitter, a social network or a news media? In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW 2010*, pages 591–600. ACM, 2010. 6

[33] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *SIGKDD 2008*, pages 462–470. ACM, 2008. 2, 6, 7, 8

[34] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *SIGKDD 2005*, pages 177–187. ACM, 2005. 2, 6, 7

[35] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3

[36] Jintang Li, Ruofan Wu, Yuchang Zhu, Huizhe Zhang, Liang Chen, and Zibin Zheng. Are large language models in-context graph learners? *CoRR*, abs/2502.13562, 2025. 4

[37] Rui Li, Jiwei Li, Jiawei Han, and Guoyin Wang. Similarity-based neighbor selection for graph llms. *CoRR*, abs/2402.03720, 2024. 4

[38] Zongzhao Li, Jiacheng Cen, Bing Su, Wenbing Huang, Tingyang Xu, Yu Rong, and Deli Zhao. Large language-geometry model: When LLM meets equivariance. *CoRR*, abs/2502.11149, 2025. 1

[39] Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. Drugchat: towards enabling chatgpt-like capabilities on drug molecule graphs. *arXiv preprint arXiv:2309.03907*, 2023. 2

[40] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 4

[41] Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. Can we soft prompt llms for graph learning tasks? In Tat-Seng Chua, Chong-Wah Ngo, Roy Ka-Wei Lee, Ravi Kumar, and Hady W. Lauw, editors, *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 481–484. ACM, 2024. 2, 4

[42] Vijay Mahajan, Eitan Muller, and Frank M Bass. New product diffusion models in marketing: A review and directions for research. *Journal of marketing*, 54(1):1–26, 1990. 6

[43] Qiheng Mao, Zemin Liu, Chenghao Liu, Zhuo Li, and Jianling Sun. Advancing graph representation learning with large language models: A comprehensive survey of techniques. *CoRR*, abs/2402.05952, 2024. 1

[44] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. 9

[45] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang, editors, *WSDM 2017*, pages 601–610. ACM, 2017. 6, 7

[46] Himchan Park and Min-Soo Kim. Evograph: An effective and efficient graph upscaling method for preserving graph properties. In Yike Guo and Faisal Farooq, editors, *KDD 2018*, pages 2051–2059. ACM, 2018. 8

[47] Yijian Qin, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Disentangled representation learning with large language models for text-attributed graphs. *CoRR*, abs/2310.18152, 2023. 4

[48] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. 3

[49] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.*, 29(1):17–37, 2017. 9

[50] Shengyin Sun, Yuxiang Ren, Chen Ma, and Xuecang Zhang. Large language models as topological structure enhancers for text-attributed graphs. *CoRR*, abs/2311.14324, 2023. 4

[51] Yuanfu Sun, Zhengnan Ma, Yi Fang, Jing Ma, and Qiaoyu Tan. Graphicl: Unlocking graph learning potential in llms through structured prompt design. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 2440–2459. Association for Computational Linguistics, 2025. 4

[52] Yanchao Tan, Hang Lv, Xinyi Huang, Jiawei Zhang, Shiping Wang, and Carl Yang. Musegraph: Graph-oriented instruction tuning of large language models for generic graph mining. *CoRR*, abs/2403.04780, 2024. 4

[53] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 491–500. ACM, 2024. 4

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[55] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of GNN representations with LLM token embeddings. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 3, 4

[56] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 4

[57] Runze Wang, Mingqi Yang, and Yanming Shen. Graph2token: Make llms understand molecule graphs. In *ICML 2024 Workshop on Efficient and Accessible Foundation Models for Biological Discovery*. 2

[58] Xiao Wang, Yiding Zhang, and Chuan Shi. Hyperbolic heterogeneous information network embedding. In *AAAI 2019*, pages 5337–5344. AAAI Press, 2019. 6, 7, 8, 9

[59] Yanbang Wang, Hejie Cui, and Jon M. Kleinberg. Microstructures and accuracy of graph recall by large language models. *CoRR*, abs/2402.11821, 2024. 4

[60] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 4

[61] Zhe Xu, Kaveh Hassani, Si Zhang, Hanqing Zeng, Michihiro Yasunaga, Limei Wang, Dongqi Fu, Ning Yao, Bo Long, and Hanghang Tong. How to make llms strong node classifiers? *arXiv preprint arXiv:2410.02296*, 2024. 3, 4

[62] Keqiang Yan, Xiner Li, Hongyi Ling, Kenna Ashen, Carl Edwards, Raymundo Arróyave, Marinka Zitnik, Heng Ji, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Invariant tokenization of crystalline materials for language model enabled generation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 1

[63] Yang Yang, Yuxiao Dong, and Nitesh V. Chawla. Microscopic evolution of social networks by triad position profile. *CoRR*, abs/1310.1525, 2013. 8

[64] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Natural language is all a graph needs. *CoRR*, abs/2308.07134, 2023. 2, 3, 4

[65] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 1955–1973. Association for Computational Linguistics, 2024. 9

[66] Hao Yin, Austin R. Benson, and Jure Leskovec. The local closure coefficient: A new perspective on network clustering. In J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman, editors, *WSDM 2019*, pages 303–311. ACM, 2019. 6, 8, 15

[67] Shuo Yu, Yingbo Wang, Ruolin Li, Guchun Liu, Yanming Shen, Shaoxiong Ji, Bowen Li, Fengling Han, Xiuzhen Zhang, and Feng Xia. Graph2text or graph2token: A perspective of large language models for graph learning. *CoRR*, abs/2501.01124, 2025. 1

[68] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR 2018*, pages 3712–3722. Computer Vision Foundation / IEEE Computer Society, 2018. 9

[69] Chengxi Zang, Peng Cui, Christos Faloutsos, and Wenwu Zhu. On power law growth of social networks. *IEEE Trans. Knowl. Data Eng.*, 30(9):1727–1740, 2018. 6

[70] Giselle Zeno, Timothy La Fond, and Jennifer Neville. Dynamic network modeling from motif-activity. In Amal El Fallah Seghrouchni, Gita Sukthankar, Tie-Yan Liu, and Maarten van Steen, editors, *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 390–397. ACM / IW3C2, 2020. 6, 8

[71] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, and Wenwu Zhu. Llm4dyg: Can large language models solve spatial-temporal problems on dynamic graphs? In Ricardo Baeza-Yates and Francesco Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 4350–4361. ACM, 2024. 4

[72] Ziwei Zhang, Haoyang Li, Zeyang Zhang, Yijian Qin, Xin Wang, and Wenwu Zhu. Large graph models: A perspective. *CoRR*, abs/2308.14522, 2023. 9

[73] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael M. Bronstein, Zhaocheng Zhu, and Jian Tang. Graphtext: Graph reasoning in text space. *CoRR*, abs/2310.01089, 2023. 4

[74] Chuang Zhou, Zhu Wang, Shengyuan Chen, Jiahe Du, Qiyuan Zheng, Zhaozhuo Xu, and Xiao Huang. Taming language models for text-attributed graph learning with decoupled aggregation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3463–3474. Association for Computational Linguistics, 2025. 3, 4

[75] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020. 1

[76] Qi Zhu, Da Zheng, Xiang Song, Shichang Zhang, Bowen Jin, Yizhou Sun, and George Karypis. Parameter-efficient tuning large language models for graph representation learning. *CoRR*, abs/2404.18271, 2024. 3, 4

[77] Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, and Yongfeng Zhang. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*, 2025. 4

A Some New Observation Spaces and Newly Discovered Graph Parameters

A.1 New Spaces

In [14], the power law is revisited based on the eigendecomposition and singular value decomposition to guide the presence of power laws in terms of the degree distribution, singular value (of adjacency matrix) distribution, and the eigenvalue (of Laplacian matrix) distribution. The authors [14] discovered that (1) degree distribution, singular value distribution, and eigenvalue distribution follow power law distribution in many real-world networks they collected; (2) a significant power law distribution of degrees usually indicates power law distributed singular values and power law distributed eigenvalues with a high probability.

A.2 New Parameters

Currently, if not all, most graph law research focuses on traditional graph properties, such as the number of nodes, the number of edges, degrees, diameters, eigenvalues, and singular values. Here, we provide some recently proposed graph properties, although they have not yet been tested on the scale for fitting the graph law on real-world networks.

The local closure coefficient [66] is defined as the fraction of length-2 paths (wedges) emanating from the head node (of the wedge) that induce a triangle, i.e., starting from a seed node of a wedge, how many wedges are closed. According to [66], features extracted within the constraints of the local closure coefficient can improve the link prediction accuracy. The local g of node u is defined as follows.

$$H(u) = \frac{2T(u)}{W^h(u)}$$

where $W^{(h)}(u)$ is the number of wedges where u stands for the head of the wedge, and $T(u)$ denotes the number of triangles that contain node u.

The density of states (or spectral density) [10] is defined as follows.

$$\mu(\lambda) = \frac{1}{N}\sum_{i=1}^{N} \delta(\lambda - \lambda_i), \int f(\lambda)\mu(\lambda) = \text{trace}(f(H)) \tag{15}$$

where H denotes any symmetric graph matrix, $\lambda_1, \ldots, \lambda_N$ denote the eigenvalues of H in the ascending order, δ stands for the Dirac delta function and f is any analytic test function.