Preprints.org

Article

Not peer-reviewed version

From Message Passing to Prompting: Rethinking Graph Learning with Large Language Models

Giancarlo Manuele [*] , Alessia Romano , Tommaso Bianchi , Lucia Ferraro

Posted Date: 22 July 2025

doi: 10.20944/preprints202507.1774.v1

Keywords: graph neural networks; large language models; structured reasoning; graph representation learning; hybrid architectures; knowledge graphs; multimodal learning; transformers; semantic graphs; neural reasoning

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

From Message Passing to Prompting: Rethinking Graph Learning with Large Language Models

Giancarlo Manuele [1,*], Alessia Romano [1], Tommaso Bianchi [2] and Lucia Ferraro [2]

[1] University of Pisa
[2] Politecnico di Milano
* Correspondence: giancarlo.manuele@unipi.it

Abstract

Graph Neural Networks (GNNs) and Large Language Models (LLMs) have each emerged as dominant paradigms in machine learning, excelling respectively in structured relational reasoning and language-based generalization. GNNs are uniquely suited for learning over graph-structured data by leveraging local connectivity and relational inductive biases, finding widespread application in domains such as chemistry, social network analysis, recommendation systems, and knowledge graphs. Meanwhile, LLMs, particularly those built upon the Transformer architecture, have demonstrated remarkable capabilities in understanding and generating human language, enabling a wide range of tasks from open-domain question answering to code synthesis and multi-step reasoning. As LLMs continue to scale and exhibit emergent abilities, their application to non-sequential data types, including graphs, has gained growing interest within the research community. This survey presents a comprehensive examination of the evolving relationship between GNNs and LLMs, analyzing how these two powerful yet fundamentally different approaches can be integrated to achieve more expressive, generalizable, and semantically rich models. We begin by establishing the mathematical foundations of GNNs and LLMs, discussing their respective architectures, learning paradigms, and representational capacities. We then explore recent advances that bridge the gap between graph and language modalities, including strategies for graph serialization, hybrid model design, and multimodal pretraining. Through an in-depth comparative analysis, we highlight the strengths and limitations of each paradigm, with special attention to scalability, interpretability, and task adaptability. A taxonomy of hybrid architectures is presented, alongside illustrative use cases in biomedical informatics, scientific discovery, recommendation, and knowledge-intensive natural language processing. The survey also identifies key challenges in representation alignment, training efficiency, benchmark design, and model robustness, and offers a forward-looking perspective on future directions. These include the development of graph-native language models, few-shot reasoning over structured data, causal inference across modalities, and the ethical deployment of dual-modality AI systems. Ultimately, we argue that the fusion of GNNs and LLMs represents a promising path toward more holistic and versatile machine learning frameworks, capable of bridging symbolic structure and linguistic understanding in ways that neither can achieve alone.

Keywords: graph neural networks; large language models; structured reasoning; graph representation learning; hybrid architectures; knowledge graphs; multimodal learning; transformers; semantic graphs; neural reasoning

1. Introduction

Graph-structured data is ubiquitous across a wide range of real-world domains, including social networks, molecular biology, transportation systems, recommendation engines, and the semantic web [1]. Traditional machine learning models, designed primarily for Euclidean data such as sequences or grids, often fail to capture the complex relational inductive biases inherent in graph data. Graph Neural

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

Networks (GNNs) have emerged as a powerful paradigm to address this challenge, providing scalable and flexible architectures that integrate node features with graph topology to learn rich node, edge, and graph-level representations [2]. Since their inception, GNNs have undergone rapid evolution, encompassing a wide spectrum of models such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), Graph Isomorphism Networks (GINs), and message-passing neural networks (MPNNs), among others. These models have consistently demonstrated strong performance across diverse tasks such as node classification, link prediction, and graph classification [3]. In parallel, the field of natural language processing (NLP) has been revolutionized by the advent of large language models (LLMs), such as GPT-3, PaLM, LLaMA, and their successors. Built upon the Transformer architecture, LLMs have exhibited unprecedented capabilities in understanding, generating, and reasoning over human language, coding tasks, and even multimodal data [4]. Their success is largely attributed to the combination of massive-scale pretraining on vast corpora and the self-attention mechanism that enables modeling long-range dependencies. As these models have scaled, they have shown surprising emergent abilities, including few-shot and zero-shot learning, prompting a paradigm shift in how general intelligence is approached in AI [5]. This juxtaposition of GNNs and LLMs raises profound questions for the future of graph representation learning [6]. On one hand, GNNs are intrinsically tailored for non-Euclidean data, with strong theoretical foundations and customized architectures for graphs. On the other hand, LLMs exhibit a generalist design with a capability to process a wide array of data types through appropriate tokenization and encoding strategies, leading to the exploration of treating graphs as sequences or textual representations, enabling the use of LLMs for graph-structured problems. This convergence brings forth new research directions: Can LLMs serve as universal approximators for graph tasks traditionally reserved for GNNs? What are the limitations of sequence-based models in capturing graph inductive biases? Conversely, can the relational reasoning of GNNs be used to enhance the capabilities of LLMs, particularly in domains requiring structured reasoning or factual consistency? Moreover, hybrid architectures and multimodal models have emerged, seeking to bridge the gap between GNNs and LLMs [7]. For instance, some approaches integrate GNNs as modules within Transformer-based architectures to provide structural priors, while others embed graphs into sequences and leverage pretrained LLMs through fine-tuning or prompting [8]. Recent advancements in instruction tuning and in-context learning further raise the possibility of using LLMs as powerful few-shot learners for graph-centric tasks, potentially obviating the need for task-specific GNNs in certain applications [9]. Additionally, knowledge graphs, long a staple of symbolic AI, have found renewed relevance as grounding tools for LLMs, offering structured factual contexts to mitigate hallucinations and improve interpretability. This survey aims to explore this rapidly evolving intersection between Graph Neural Networks and Large Language Models [10]. We provide a comprehensive overview of the fundamental concepts of GNNs and LLMs, examine their respective strengths and limitations, and delve into recent efforts that integrate the two paradigms. We categorize and analyze the methods that leverage LLMs for graph learning, including textualization of graphs, graph-aware pretraining, and graph prompting. Furthermore, we explore how GNNs can be used to complement LLMs in structured reasoning, information retrieval, and knowledge integration. Through this survey, we also identify open challenges and opportunities, such as scalability, explainability, and the alignment of inductive biases, which remain critical to realizing the full potential of this convergence [11]. In the age of LLMs, where generalist models are becoming increasingly dominant, the question is not whether GNNs will be replaced, but rather how they will adapt, evolve, and integrate [12]. As we navigate this new frontier of graph learning with foundation models, this survey endeavors to provide a foundational resource for researchers and practitioners alike, offering insights, taxonomies, and perspectives that illuminate the path forward [13].

2. Preliminaries

In this section, we introduce the formal foundations of Graph Neural Networks (GNNs) and Large Language Models (LLMs), with a focus on the mathematical formulations that underpin their

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

architectures and learning paradigms [14]. We begin by defining the basic notation for graphs and proceed to discuss the general message passing framework for GNNs. We then provide a brief overview of the Transformer architecture that forms the basis of modern LLMs, highlighting key components such as self-attention, positional encoding, and large-scale pretraining.

2.1. Graph Representations and Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V}$ is the set of $n = |\mathcal{V}|$ nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Each node $v_i \in \mathcal{V}$ may be associated with a feature vector $\mathbf{x}_i \in \mathbb{R}^d$, and similarly, edges $(v_i, v_j) \in \mathcal{E}$ may have associated features $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$ [15]. The graph can be represented by its adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, where $A_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and 0 otherwise [16]. In many cases, the adjacency matrix is extended to a weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ to model edge weights [17]. A key objective in graph learning is to learn a function f_θ parameterized by θ that maps the input graph (and its features) to desired outputs [18]. This could be node-level predictions $f_\theta : \mathcal{V} \to \mathbb{R}^k$, edge-level predictions $f_\theta : \mathcal{E} \to \mathbb{R}^k$, or whole-graph embeddings $f_\theta : \mathcal{G} \to \mathbb{R}^k$, depending on the task [19].

2.2. Message Passing Neural Networks

The predominant formulation of GNNs is the Message Passing Neural Network (MPNN) framework, which generalizes many GNN architectures [20]. Given node features $\{\mathbf{h}_i^{(0)} = \mathbf{x}_i\}_{i=1}^n$, a typical GNN performs T rounds of message passing, where in each layer $t \in \{1, \ldots, T\}$, node embeddings are updated by aggregating messages from neighbors. The general update rule can be written as:

$$\mathbf{m}_i^{(t)} = \text{AGGREGATE}^{(t)}\left(\left\{\phi_m^{(t)}\left(\mathbf{h}_i^{(t-1)}, \mathbf{h}_j^{(t-1)}, \mathbf{e}_{ij}\right) : j \in \mathcal{N}(i)\right\}\right)$$

$$\mathbf{h}_i^{(t)} = \phi_u^{(t)}\left(\mathbf{h}_i^{(t-1)}, \mathbf{m}_i^{(t)}\right)$$

Here, $\mathcal{N}(i)$ denotes the neighborhood of node i, $\phi_m^{(t)}$ is a message function that may depend on node and edge features, and $\phi_u^{(t)}$ is an update function such as an MLP. The AGGREGATE function is typically a permutation-invariant operator such as summation, mean, or max. The final node embeddings $\mathbf{h}_i^{(T)}$ can be used directly for node-level tasks or pooled via a READOUT function (e.g., sum or attention) for graph-level predictions. Different GNN variants correspond to different instantiations of ϕ_m, ϕ_u, and AGGREGATE [21]. For example, Graph Convolutional Networks (GCNs) use a normalized sum aggregation with linear updates, while Graph Attention Networks (GATs) compute attention-weighted sums using learned coefficients. Graph Isomorphism Networks (GINs) adopt a sum aggregation with injective MLPs, enhancing expressivity in distinguishing graph structures.

2.3. Transformer-Based Language Models

Modern LLMs such as GPT and BERT are based on the Transformer architecture, which eschews recurrence in favor of self-attention mechanisms. A Transformer processes a sequence of tokens $\mathbf{x} = (x_1, x_2, \ldots, x_L)$, where each token is mapped to an embedding $\mathbf{e}_i \in \mathbb{R}^d$ [22]. The core building block is the multi-head self-attention mechanism, which computes interactions between all pairs of tokens in the sequence. Given input embeddings $\mathbf{X} \in \mathbb{R}^{L \times d}$, the self-attention mechanism computes:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}^K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}^V$ are the query, key, and value matrices obtained via learned projections. The result is a new representation of the sequence where each token attends to all others, modulated by similarity in the query-key space [23]. Multiple attention heads are used in parallel, and their outputs are concatenated and passed through a feed-forward network (FFN) [24]. Each Transformer block also includes layer normalization and residual connections [25]. Crucially, positional encodings $\mathbf{P} \in \mathbb{R}^{L \times d}$ are added to $\mathbf{X}$ to preserve order information, since self-attention is

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

inherently permutation-invariant [26]. LLMs are typically trained on large corpora via autoregressive (e.g., GPT) or masked language modeling (e.g., BERT) objectives. Given their scale and generality, they acquire broad world knowledge and can generalize to diverse tasks with little or no fine-tuning, a capability enhanced by prompt-based or instruction-tuned approaches.

2.4. Comparative Representational Capabilities

While GNNs are naturally suited for graph-structured data, their expressive power is bounded by the Weisfeiler-Lehman (WL) test, especially in standard message-passing formulations [27]. Recent work has attempted to surpass this barrier using higher-order or subgraph-based methods. On the other hand, LLMs have shown strong capacity for symbolic reasoning and relational inference despite being trained on sequences, raising intriguing questions about their ability to model graphs [28]. A central theoretical question is the representational alignment between GNNs and LLMs: can Transformers implicitly simulate GNN-style computations over serialized graph data [29]? Conversely, can GNNs encode linguistic or hierarchical structure that is typically handled by LLMs? Addressing these questions requires both empirical benchmarking and theoretical insight, which we explore in subsequent sections [30].

3. Comparative Analysis of GNNs and LLMs for Graph-Based Tasks

As the capabilities of Large Language Models (LLMs) continue to expand, their application to traditionally non-sequential domains—such as graphs—has become an increasingly compelling research direction [31]. In this section, we provide a comparative analysis of Graph Neural Networks (GNNs) and LLMs with respect to their performance, scalability, expressivity, and inductive biases in various graph-centric tasks [32]. Our analysis is grounded in both theoretical considerations and empirical results drawn from recent literature. We also include a summarizing table that contrasts these two paradigms across multiple dimensions, with an emphasis on their relative strengths and limitations [33].

Table 1. Comparison of Graph Neural Networks (GNNs) and Large Language Models (LLMs) for Graph-Based Learning.

Aspect	Graph Neural Networks (GNNs)	Large Language Models (LLMs)
Input Representation	Native graph structure (adjacency matrix, edge lists)	Sequentialized or textual graph encodings (e.g., node triples, graph prompts)
Architecture Bias	Local and relational inductive bias via message passing	Global attention with soft inductive bias to sequence and co-occurrence statistics
Scalability	Efficient for sparse and localized graphs; limited by neighborhood explosion in deep GNNs	Highly scalable with sufficient hardware; cost grows quadratically with input length
Expressivity	Limited by the 1-WL test; extensions include higher-order GNNs	Capable of learning high-order dependencies; lacks explicit graph inductive priors
Pretraining	Domain-specific pretraining schemes (e.g., node masking, graph contrastive learning)	Trained on massive text corpora; general-purpose representations can be repurposed
Adaptability to Tasks	Requires task-specific architecture design and training	Zero-shot and few-shot learning enabled by prompting and in-context learning
Explainability	More interpretable due to structured graph flow and neighborhood influence	Difficult to interpret due to dense attention and token-level reasoning
Integration with External Knowledge	Can directly incorporate structured knowledge graphs as input	Often needs grounding via knowledge injection, retrieval, or augmented prompts
Fine-tuning Overhead	Lightweight and efficient on small datasets	Expensive and often impractical without parameter-efficient methods

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

The dichotomy between GNNs and LLMs is evident in their respective design philosophies [34]. GNNs are intrinsically built to respect the relational structure of graphs, leveraging edge connectivity and localized message passing to build representations. This design results in models that are compact, efficient for sparse graphs, and inherently interpretable with regard to the graph topology. However, traditional GNNs are limited in expressivity, typically bounded by the discriminative power of the 1-WL test, and they often require custom architecture modifications to handle specific graph types (e.g., heterogeneous, dynamic, or hypergraphs) [35]. Conversely, LLMs are fundamentally sequence-based models [36]. Their lack of native support for graphs is offset by their impressive generalization capacity, which allows them to operate over serialized representations of graphs when carefully formatted [37]. For instance, node-link triplets, adjacency lists, and traversal paths can be converted into textual prompts that an LLM can process [38]. While this allows LLMs to perform surprisingly well in some graph-related tasks—such as question answering over knowledge graphs or predicting properties of molecular graphs—it introduces substantial challenges. The flattening of structure into sequences can obscure important topological cues, and token-based attention mechanisms may struggle with long-range or high-degree dependencies [39]. An area where LLMs show clear promise is in their adaptability [40]. With instruction tuning, prompt engineering, and few-shot in-context learning, LLMs can be rapidly adapted to new graph-based tasks without modifying the model parameters. This is in stark contrast to GNNs, which typically require full retraining and architectural customization for each task. However, LLMs come with heavy computational overhead and memory requirements, especially when dealing with long input sequences derived from large or dense graphs [41]. Another critical dimension is interpretability [42]. GNNs offer relatively transparent computation paths, where node embeddings can often be traced back to local neighborhoods. Techniques such as subgraph extraction, attention visualization, and influence functions provide further interpretability tools. LLMs, however, are notoriously opaque, with dense and entangled attention matrices making it difficult to identify causal inputs or reasoning chains, especially when graph structure is implicitly encoded. Finally, integration with external knowledge sources presents both opportunities and challenges for both paradigms. GNNs can directly operate over knowledge graphs, using their structure to guide message passing [43]. LLMs often require grounding through retrieval-augmented generation or explicit prompt engineering to align with structured knowledge. Despite this, LLMs can generalize from textual patterns in ways that GNNs cannot, offering robustness in open-domain and noisy scenarios. In light of these differences, it becomes increasingly apparent that the two paradigms are not mutually exclusive but instead offer complementary strengths. Hybrid models that integrate the structural inductive biases of GNNs with the semantic and generalization power of LLMs represent a promising frontier, which we explore in detail in the next section.

4. Hybrid Architectures: Bridging GNNs and LLMs

The complementary strengths of Graph Neural Networks (GNNs) and Large Language Models (LLMs) have inspired a growing body of work exploring hybrid architectures that integrate the structural inductive biases of GNNs with the powerful representation and reasoning abilities of LLMs [44]. These hybrid models are motivated by the observation that many real-world problems—such as question answering over knowledge bases, scientific reasoning, recommendation systems, and drug discovery—require both the explicit modeling of relational structure and the ability to interpret or generate natural language. In this section, we explore several classes of such hybrid architectures and present a unifying view of their interaction patterns. At a high level, hybrid architectures can be categorized into three main paradigms: (i) GNN-enhanced LLMs, where structural information is injected into LLMs through graph embeddings or structured prompts; (ii) LLM-enhanced GNNs, where pretrained LLMs provide semantic or contextual features for nodes or edges; and (iii) interleaved models, where GNN and LLM components interact iteratively or in tandem during a task. These designs attempt to combine the localized, interpretable reasoning of GNNs with the global, context-aware understanding of LLMs [45].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

As illustrated in Figure 1, one common strategy is to use a GNN to first process the raw graph data and generate node or graph-level embeddings. These embeddings are then linearized or embedded into a sequence format (e.g., via positional concatenation or structured templates) that can be ingested by an LLM [46]. The LLM, in turn, may use its language modeling capabilities to reason over the embeddings, complete textual tasks, or perform structured generation. This approach has been used effectively in molecular property prediction, where GNNs model the graph of atoms and bonds, and the LLM generates textual rationales or interprets properties in natural language. In a reversed configuration, LLMs can act as contextual encoders for node or edge features [47]. For example, in knowledge graphs or citation networks, textual descriptions or abstracts associated with nodes can be encoded by an LLM and used as input features to a GNN [48]. The GNN then processes the graph structure using these rich embeddings, enabling it to combine semantic context with relational patterns [49]. This strategy has proven effective in low-resource and few-shot settings, where pretraining from LLMs compensates for limited graph data. A more recent and sophisticated line of research explores interleaved or co-trained models, where LLM and GNN components communicate iteratively. For instance, in multi-hop question answering, an LLM may generate candidate paths or node queries, which are then validated or reranked by a GNN operating on a knowledge graph. Conversely, the GNN may identify subgraphs relevant to a query, and the LLM can be prompted with serialized versions of these subgraphs to generate answers [50]. These bidirectional systems aim to leverage the strengths of both components in a tightly coupled manner, often leading to state-of-the-art performance in structured reasoning tasks. Despite their promise, hybrid architectures face several practical challenges. The serialization of graph embeddings into textual or sequence form can lead to information loss, especially for large or densely connected graphs. Additionally, training such systems end-to-end can be computationally demanding, and the fusion of disparate architectures introduces complex design choices regarding attention mechanisms, representation alignment, and loss coordination [51]. Nonetheless, early successes suggest that the combination of GNNs and LLMs can be greater than the sum of their parts, particularly in domains where structured data and natural language intersect. The exploration of these hybrid models not only opens new application frontiers but also provokes deeper questions about representation unification and modality translation [52]. As we progress toward more general-purpose, multimodal AI systems, understanding and refining the interfaces between graph-based and sequence-based learning will become increasingly vital [53].

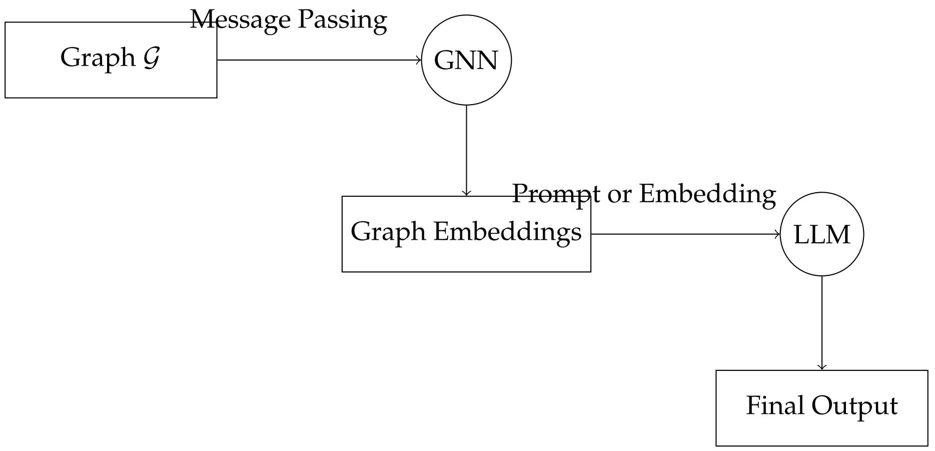

Figure 1. Illustration of a simple hybrid architecture: a GNN processes a graph to produce embeddings, which are then fed into an LLM for downstream reasoning or generation tasks.

5. Applications in the Wild

The convergence of Graph Neural Networks (GNNs) and Large Language Models (LLMs) has not only spurred theoretical advancements but also led to impactful applications across a diverse set of real-world domains [54]. These hybrid and cross-paradigm approaches have shown remarkable potential in tasks where both structured graph reasoning and semantic understanding are critical. In

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

this section, we explore how the joint capabilities of GNNs and LLMs are being utilized in fields such as biomedical informatics, scientific discovery, recommendation systems, natural language processing, and knowledge-intensive tasks [55]. In biomedical domains, molecular property prediction, drug discovery, and protein interface modeling are inherently graph-based problems [56]. Molecules are naturally represented as graphs of atoms connected by bonds, and GNNs have been highly effective at capturing such structural information. However, domain-specific knowledge embedded in chemical nomenclature, scientific publications, or reaction protocols is not easily encoded in the graph alone. LLMs, particularly those trained on biomedical corpora (e.g., PubMed, ChEMBL abstracts), provide semantic context that complements graph-based features [57]. For instance, a GNN might identify a substructure within a molecule as potentially reactive, while an LLM could explain its function or relevance based on known pharmacological patterns [58]. Recent models such as GraphGPT and BioMedLM-GNN hybrids are emerging to jointly exploit symbolic and topological signals, enabling more interpretable and data-efficient discovery pipelines. In the scientific research domain more broadly, graph-based citation networks and co-authorship graphs are commonly used to model scholarly communication. Traditional GNNs have been used to predict missing citations, identify influential papers, and cluster research communities. However, combining these relational insights with LLM-generated textual embeddings of paper abstracts, titles, and keywords enables a more nuanced understanding of topic evolution, novelty, and scientific impact. Hybrid architectures that use GNNs to navigate the citation topology while leveraging LLMs to interpret textual content are being deployed in intelligent literature review tools and AI-based research assistants, aiding both information retrieval and hypothesis generation. In recommendation systems, user-item interactions can be naturally modeled as bipartite graphs, which GNNs have used extensively to propagate preferences and uncover collaborative filtering patterns [59]. However, many recommendation scenarios also involve textual descriptions—such as product reviews, movie summaries, or news articles—which encode latent semantic signals that are not explicitly modeled in the graph [60]. By incorporating LLM-derived embeddings from such textual sources into the node representations within a GNN, or by prompting LLMs with graph-based context, systems can better understand user intent and item semantics. This has led to improvements not only in recommendation accuracy but also in explainability, as LLMs can be prompted to provide natural language rationales for suggestions [61]. In knowledge-intensive natural language processing tasks—such as question answering, fact verification, and information extraction—knowledge graphs play a central role in grounding language in structured reality. GNNs have traditionally served as the reasoning engine over such graphs, while LLMs functioned as the language interface [62]. Recent trends, however, move toward a deeper integration, where LLMs are prompted with serialized subgraphs, and their outputs are fed back into GNN modules for further structural validation or disambiguation [63]. Systems like Retrieval-Augmented Generation (RAG) and Knowledge-Aware Transformers leverage this tight coupling to mitigate hallucinations, improve factual consistency, and facilitate multi-hop reasoning across entities and relations. Moreover, in domains such as legal or financial analytics, where knowledge bases are critical but often noisy, GNNs can denoise or reweight evidence sources, while LLMs synthesize natural language answers that preserve logical coherence. Even in the domain of software engineering and program analysis, the synergy between GNNs and LLMs is gaining traction. Abstract syntax trees (ASTs), control flow graphs (CFGs), and call graphs are used to model code structure, and GNNs have proven valuable for tasks like vulnerability detection, code summarization, and type inference [64]. Meanwhile, LLMs such as Codex and CodeBERT are proficient in generating and understanding source code from textual prompts. By combining the two—using GNNs to reason about structural dependencies in code and LLMs to handle comments, documentation, or naming conventions—modern AI-powered development tools are achieving unprecedented accuracy and generalizability in code understanding tasks [65]. In all these domains, the integration of GNNs and LLMs leads to models that are not only more performant but also more robust and interpretable [66]. This is especially crucial in high-stakes or low-resource environments where generalization,

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

transparency, and data efficiency are paramount. Importantly, these applications also reveal a larger paradigm shift: as data becomes increasingly multimodal and interconnected, the need for models that can fluidly move between graphs and language—structure and semantics—becomes both inevitable and indispensable.

6. Challenges and Open Problems

Despite the rapid progress and impressive results achieved at the intersection of Graph Neural Networks (GNNs) and Large Language Models (LLMs), a number of critical challenges remain unresolved [67]. These challenges are both theoretical and practical, spanning issues of scalability, representation alignment, task generalization, evaluation, and interpretability [68]. Addressing these open problems is vital not only for refining current hybrid approaches but also for paving the way toward a unified theory of graph and language learning in machine intelligence. One of the foremost challenges lies in the representation gap between graphs and sequences [69]. GNNs operate natively on structured, non-Euclidean data, where topology and connectivity define the primary inductive biases [70]. In contrast, LLMs consume flat, linearized sequences of tokens, and their attention mechanisms are optimized for positional and co-occurrence statistics [71]. When graphs are serialized into sequences—using techniques such as random walks, adjacency encodings, or prompt-based node tuples—much of the relational information may be distorted or lost. The challenge, then, is to develop more faithful serialization strategies that preserve graph topology while remaining compatible with LLM tokenization schemes. Conversely, when LLM-generated outputs (e.g., reasoning traces, graph edits, or textual predictions) are mapped back into structured formats, the inverse problem of deserialization becomes equally nontrivial. This bidirectional representational bottleneck is currently a major roadblock to seamless integration [72]. Scalability is another pressing concern [73]. GNNs, particularly message-passing variants, can suffer from the well-known issues of over-smoothing and neighborhood explosion as depth increases, making them less effective on large or densely connected graphs [74]. On the other hand, LLMs are computationally intensive due to their quadratic attention complexity with respect to sequence length, which becomes prohibitive when processing serialized forms of large graphs. Hybrid models inherit the computational burdens of both paradigms, raising significant challenges in terms of memory usage, training time, and inference latency [75,76]. Efficient architectural designs, such as sparse attention mechanisms, hierarchical graph encodings, or parameter-efficient fine-tuning (e.g., LoRA, adapters) are active areas of research aimed at mitigating these scalability issues. Another open problem concerns task alignment and generalization [77]. While LLMs excel at zero-shot and few-shot generalization in language domains, their performance on graph-related tasks often hinges on the quality of the prompt or the fidelity of the graph serialization [78]. GNNs, in contrast, require full supervision or carefully designed pretraining schemes but generalize poorly to unseen graphs or tasks with distributional shift. Designing training objectives and curriculum strategies that allow hybrid models to generalize effectively—especially in low-resource or cross-domain settings—remains an unresolved challenge [79]. Furthermore, benchmarks that fairly compare GNNs, LLMs, and hybrid models on the same tasks are limited, making it difficult to draw reliable conclusions about their relative merits or synergies [80]. Interpretability and explainability are also critical issues, particularly for applications in high-stakes domains such as healthcare, finance, and law [81]. While GNNs offer some degree of interpretability via graph attention weights, influence functions, or subgraph importance scores, LLMs remain largely opaque. When combined in hybrid systems, the interpretability problem compounds: the interaction between structural and semantic reasoning becomes difficult to trace, and debugging model errors becomes more complex [82]. There is a growing need for integrated explanation frameworks that can jointly visualize and rationalize both graph-level and language-level components of model behavior. Such frameworks should ideally allow users to inspect the propagation of information across graph edges, the influence of specific tokens or prompts, and the interaction between the two in downstream predictions. Moreover, data availability and standardization pose practical challenges [83]. There is

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

a lack of large-scale, high-quality datasets that natively integrate graph structure with rich textual descriptions in a task-driven format. Most benchmarks either focus on pure graph tasks (e.g., OGB datasets) or purely textual tasks (e.g., GLUE, SuperGLUE, MMLU), with very few straddling the boundary. Synthetic datasets or heuristically constructed graph-text corpora offer a starting point but often fail to reflect the complexity of real-world graph-linguistic interactions [84]. The community urgently needs standardized benchmarks, evaluation metrics, and shared tasks that test the full pipeline of graph-to-text and text-to-graph reasoning under realistic conditions. Finally, there are foundational questions about the theoretical limits and synergies of GNNs and LLMs. To what extent can Transformers simulate graph algorithms over serialized inputs? Can message passing in GNNs be augmented with learned attention from LLMs in a theoretically grounded way? What classes of functions are expressible only through their combination? These questions touch on the nature of representation, reasoning, and learning, and answering them may require new insights from graph theory, information theory, and computational complexity. As research continues to push the boundaries of hybrid learning systems, such foundational understanding will be crucial in guiding principled model design rather than relying solely on empirical heuristics [85]. In sum, while the fusion of GNNs and LLMs offers immense promise, it is still a nascent field grappling with core scientific and engineering challenges. Progress will require coordinated efforts across algorithm design, systems optimization, theoretical analysis, and benchmark development. Addressing these challenges will not only lead to better models but also to deeper insights into how structure and semantics can be unified in intelligent systems [86].

7. Future Directions

As the boundaries between structured and unstructured data continue to blur, the interplay between Graph Neural Networks (GNNs) and Large Language Models (LLMs) is poised to redefine the next generation of machine learning systems [87]. While existing research has already demonstrated the potential of hybrid models that combine graph reasoning with language understanding, the trajectory ahead is rich with possibilities for methodological innovation, theoretical integration, and real-world impact. In this section, we outline several promising future directions that will shape the landscape of graph-language modeling in the years to come [88]. One major direction is the development of *graph-native LLMs*—models that are pretrained directly on graph-structured data rather than on sequences derived from text alone [89]. Such models would use message passing, attention over graph neighborhoods, or novel positional encodings tailored for non-Euclidean data as their fundamental building blocks. While some early works have attempted to adapt Transformers for graphs (e.g., Graphormer, SAN), there remains significant room to design pretraining objectives that capture both topological patterns and semantic relationships simultaneously [90]. These models could learn from massive knowledge graphs, citation networks, or molecular structures, potentially uncovering universal graph-language representations that generalize across domains and tasks [91]. Closely related is the idea of *multimodal graph-language pretraining*, where models are trained on datasets that pair graph structures with natural language annotations. Consider, for example, a molecular graph linked to its SMILES string and natural language description of its pharmacological properties, or a knowledge graph where each entity is accompanied by a textual description [92]. Pretraining across both modalities could enable models to translate seamlessly between graphs and text, producing graph explanations in natural language, or generating synthetic graphs from textual specifications. This direction opens the door to novel applications in explainable AI, data synthesis, and cross-modal retrieval, all of which benefit from a deep, bidirectional understanding of structure and language [93]. Another promising area is the integration of *retrieval-augmented mechanisms in graph reasoning*. While retrieval-augmented generation (RAG) has become a standard technique in LLMs for grounding open-domain question answering, similar ideas could be applied in the graph context [94]. A hybrid model could retrieve relevant subgraphs from a large corpus of knowledge graphs or dynamically construct local graphs based on textual queries, then use GNNs to perform reasoning and LLMs to

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

interpret or refine the outputs [95]. This dynamic interplay between retrieval, graph construction, and linguistic inference could enable highly flexible and efficient systems for tasks such as fact verification, legal reasoning, or semantic search [96]. *Few-shot and zero-shot generalization over graphs* is another frontier of considerable importance [97]. While LLMs have shown remarkable performance in few-shot learning through in-context examples, extending this capability to graph domains is nontrivial. Future work could investigate how few-shot prompts, combined with structural priors from GNNs, might enable models to generalize to unseen graphs, new relations, or novel node types with minimal supervision. Techniques such as adapter layers, soft prompting, or meta-learning could help bridge the inductive gaps between language and structure, creating more versatile and sample-efficient learners. A significant research opportunity also lies in the realm of *causal reasoning over graph-text interfaces* [98]. Many real-world problems—such as disease progression modeling, financial risk assessment, or policy analysis—involve not just correlation but causal inference across structured and textual data. Future models may need to reason about interventions, counterfactuals, or temporal dependencies across graphs (e.g., patient treatment networks) while synthesizing explanations in natural language [99]. This will require a synthesis of graph-theoretic causal modeling with the generative reasoning abilities of LLMs, along with new training objectives and evaluation criteria grounded in causal semantics [100]. Finally, as these models become increasingly powerful and autonomous, questions of *robustness, fairness, and ethical alignment* will become ever more pressing [101]. Graphs are susceptible to adversarial attacks, and LLMs are known to hallucinate or propagate biases from their training data [102]. The intersection of the two paradigms introduces compounded risks—for example, generating biased or misleading textual justifications for incorrect graph inferences. Future research must address these concerns by developing interpretable and verifiable hybrid systems, establishing safeguards against misuse, and exploring governance frameworks for deploying such models in socially sensitive contexts [103]. In conclusion, the fusion of GNNs and LLMs represents more than just an architectural trend—it signals a deeper shift toward AI systems that can reason fluidly across structure and language [104]. The challenges ahead are substantial, but so are the rewards: more intelligent, generalizable, and human-aligned models that can understand the world not just as a sequence of words or a set of edges, but as a rich interplay of structure, semantics, and meaning. As this field matures, it is likely to produce foundational advances that resonate across disciplines, from natural science and engineering to education, healthcare, and beyond.

8. Conclusion

The rapid co-evolution of Graph Neural Networks (GNNs) and Large Language Models (LLMs) marks a transformative period in the landscape of machine learning, one in which structure-aware reasoning and language-based understanding are no longer pursued in isolation. This survey has explored the conceptual foundations, architectural strategies, practical applications, and open challenges that emerge at the intersection of these two powerful paradigms. While GNNs offer precise, topology-sensitive modeling of relational data, LLMs bring an unprecedented capacity for flexible reasoning, semantic abstraction, and few-shot generalization. Together, they form the building blocks of a new class of models capable of unifying structure and language into cohesive, intelligent systems.

We have seen that GNNs excel at learning from graph-structured inputs by exploiting the locality and inductive biases of node connectivity. They offer interpretable paths through which information flows from neighbors and have demonstrated strong empirical performance across a variety of domains including chemistry, recommendation systems, and social network analysis. Yet, their limitations in generalization, representation power (e.g., the 1-WL bottleneck), and reliance on labeled data often constrain their scalability and flexibility. Conversely, LLMs have emerged as general-purpose learners, capable of synthesizing, generating, and reasoning with minimal supervision. Their strength lies in semantic fluency and context modeling, but they are fundamentally disadvantaged when it comes to explicitly handling structured, graph-based relationships—unless such structure is carefully encoded into textual form.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

The hybridization of these approaches is not merely additive but potentially synergistic. By incorporating graph embeddings into LLM prompts or using LLMs to contextualize node and edge features in GNNs, researchers have begun to unlock capabilities that neither paradigm can achieve alone. These models have already begun to influence critical tasks such as molecular generation, knowledge graph completion, scientific literature synthesis, and multi-hop reasoning. The promise lies not just in task performance, but in the possibility of creating models that reason more like humans—by fluidly navigating both abstract structures and rich semantics.

Despite these advances, it is clear that much work remains. Fundamental gaps persist in how graphs and sequences interact within neural architectures, and existing models are often inefficient, hard to interpret, or brittle in the face of distributional shifts. There are also philosophical and theoretical questions yet to be fully addressed: Can one architecture ultimately subsume the capabilities of the other? Or will hybrid models always be necessary to strike the right balance between relational and linguistic generalization? Furthermore, as models continue to scale and be deployed in high-stakes environments, ensuring robustness, fairness, and alignment with human values will be critical.

Ultimately, this survey underscores that GNNs and LLMs are not competing paradigms, but rather complementary perspectives on intelligence. Just as graphs offer a powerful abstraction for structure, language provides a universal interface for communication and reasoning. Bridging the two allows for the development of models that are not only more powerful and general, but also more aligned with the way we as humans understand and navigate the world. The next generation of AI will likely be built not by choosing between graphs or language, but by learning how to harness both in concert, leveraging their unique strengths to tackle increasingly complex, multimodal, and interconnected problems. As research at this intersection continues to mature, it holds the potential to fundamentally reshape how we model, reason about, and interact with structured knowledge in the age of large-scale machine learning.

References

1. N. M. Kriege. *Comparing Graphs: Algorithms & Applications*. Phd thesis, TU Dortmund University, 2015.
2. Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *ICLR*, pages 321–328, 2003.
3. Christos Louizos, Uri Shalit, Joris M. Mooij, David A. Sontag, Richard S. Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *NeurIPS*, 2017.
4. A. Schrijver. *Theory of Linear and Integer programming*. 1986.
5. Yixuan He, Xitong Zhang, Junjie Huang, Benedek Rozemberczki, Mihai Cucuringu, and Gesine Reinert. Pytorch geometric signed directed: a software package on graph neural networks for signed and directed graphs. In *Learning on Graphs Conference*, pages 12–1. PMLR, 2024.
6. Lin Ni, Sijie Wang, Zeyu Zhang, Xiaoxuan Li, Xianda Zheng, Paul Denny, and Jiamou Liu. Enhancing student performance prediction on learnersourced questions with sgnn-llm synergy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23232–23240, 2024.
7. Chuizheng Meng, Sam Griesemer, Defu Cao, Sungyong Seo, and Yan Liu. When physics meets machine learning: A survey of physics-informed machine learning. *Machine Learning for Computational Science and Engineering*, 1(1):1–23, 2025.
8. Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy S Liang, and Jure Leskovec. Prodigy: Enabling in-context learning over graphs. *Advances in Neural Information Processing Systems*, 36: 16302–16317, 2023.
9. U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International journal on document analysis and recognition*, 5:39–46, 2002.
10. Christopher Morris, Gaurav Rattan, Sandra Kiefer, and Siamak Ravanbakhsh. SpeqNets: Sparsity-aware permutation-equivariant graph networks. In *ICLR*, pages 16017–16042, 2022.
11. Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Massachusetts, 1985.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

12. Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 974–983, 2018.
13. Giovanni Da San Martino, Nicolò Navarin, and Alessandro Sperduti. A tree-based kernel for graphs. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012*, pages 975–986, 2012.
14. Alvaro Arroyo, Bruno Scalzo, Ljubiša Stanković, and Danilo P Mandic. Dynamic portfolio cuts: A spectral approach to graph-theoretic diversification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5468–5472. IEEE, 2022.
15. Filip Ekström Kelvinius, Dimitar Georgiev, Artur Toshev, and Johannes Gasteiger. Accelerating molecular graph neural networks via knowledge distillation. *Advances in Neural Information Processing Systems*, 36: 25761–25792, 2023.
16. Gal Vardi. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66(6):86–93, 2023.
17. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
18. Christopher Morris, Floris Geerts, Jan Tönshoff, and Martin Grohe. WL meet VC. In *ICML*, 2023.
19. Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2021.
20. M. Grohe, P. Schweitzer, and D. Wiebking. Deep Weisfeiler Leman. *arXiv preprint*, 2020.
21. Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International conference on machine learning*, pages 3469–3489. PMLR, 2022.
22. Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami flow and neural diffusion on graphs. *Advances in Neural Information Processing Systems*, 34:1594–1609, 2021.
23. Tao Lei, Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Deriving neural architectures from sequence and graph kernels. In *ICLR*, pages 2024–2033, 2017.
24. J. Klicpera, F. Becker, and S. Günnemann. Gemnet: Universal directional graph neural networks for molecules. *arXiv preprint*, 2021.
25. Simon S. Du, Kangcheng Hou, Ruslan Salakhutdinov, Barnabás Póczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *NeurIPS 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5724–5734, 2019.
26. Tianjin Huang, Tianlong Chen, Meng Fang, Vlado Menkovski, Jiaxu Zhao, Lu Yin, Yulong Pei, Decebal Constantin Mocanu, Zhangyang Wang, Mykola Pechenizkiy, and Shiwei Liu. You can have better graph neural networks by not training weights at all: Finding untrained GNNs tickets. In *LoG*, 2022.
27. Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak A. Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *ICLR*, 2019.
28. 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. *arXiv preprint arXiv:2305.19523*, 2023.
29. Francesco Di Giovanni, T Konstantin Rusch, Michael M Bronstein, Andreea Deac, Marc Lackenby, Siddhartha Mishra, and Petar Veličković. How does over-squashing affect the power of gnns? *arXiv:2306.03589*, 2023.
30. Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. *arXiv preprint arXiv:2203.00199*, 2022.
31. Beatrice Bevilacqua, Moshe Eliasof, Eli A. Meirom, Bruno Ribeiro, and Haggai Maron. Efficient subgraph GNNs by learning effective selection policies. *arXiv preprint*, abs/2310.20082, 2023.
32. V. M. Zolotarev. *One-dimensional stable distributions*. Providence, RI, 1986.
33. T. Gärtner. *Kernels for Structured Data*. PhD thesis, University of Bonn, 2005.
34. Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *ICLR*, 2020.
35. Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. Graph message passing with cross-location attentions for long-term ili prediction. *arXiv preprint arXiv:1912.10202*, 2019.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

36. Andrea Cini, Daniele Zambon, and Cesare Alippi. Sparse graph learning from spatiotemporal time series. *Journal of Machine Learning Research*, 24(242):1–36, 2023.
37. He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. Trustworthy graph neural networks: Aspects, methods and trends. *arXiv preprint arXiv:2205.07424*, 2022.
38. Ben Day, Cătălina Cangea, Arian R Jamasb, and Pietro Liò. Message passing neural processes. *arXiv preprint arXiv:2009.13895*, 2020.
39. Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning*, pages 2528–2547. PMLR, 2023.
40. N. Kriege, M. Neumann, K. Kersting, and M. Mutzel. Explicit versus implicit graph feature maps: A computational phase transition for walk kernels. In *IEEE International Conference on Data Mining*, pages 881–886, 2014.
41. Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *ICLR*, pages 321–328, 2003.
42. Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.
43. Fredrik D. Johansson, Vinay Jethava, Devdatt P. Dubhashi, and Chiranjib Bhattacharyya. Global graph kernels using geometric embeddings. In *ICLR*, pages 694–702, 2014.
44. Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. Revisiting over-smoothing in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020.
45. M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, Z. Huang, Q. Guo, H. Zhang, H. Lin, J. Zhao, J. Li, A. J. Smola, and Z. Zhang. Deep graph library: Towards efficient and scalable deep learning on graphs. *arXiv preprint*, 2019.
46. Zaïd Harchaoui and Francis R. Bach. Image classification with segmentation graph kernels. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.
47. Alan Thomas, Robert Gaizauskas, and Haiping Lu. Leveraging llms for post-ocr correction of historical newspapers. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA)@ LREC-COLING-2024*, pages 116–121, 2024.
48. Francesco Di Giovanni, T. Konstantin Rusch, Michael Bronstein, Andreea Deac, Marc Lackenby, Siddhartha Mishra, and Petar Veličković. How does over-squashing affect the power of GNNs? *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.
49. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.
50. Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of Gnns via graph biconnectivity. *arXiv preprint*, 2023.
51. Liang Yang, Weihang Peng, Wenmiao Zhou, Bingxin Niu, Junhua Gu, Chuan Wang, Yuanfang Guo, Dongxiao He, and Xiaochun Cao. Difference residual graph neural networks. In *Proceedings of the 30th ACM international conference on multimedia*, pages 3356–3364, 2022.
52. Matteo Togninalli, M. Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten M. Borgwardt. Wasserstein weisfeiler-lehman graph kernels. In *NeurIPS 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6436–6446, 2019.
53. Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *ICLR*, 2021.
54. Tao Lei, Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Deriving neural architectures from sequence and graph kernels. In *ICLR*, pages 2024–2033, 2017.
55. D. Rogers and M. Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, (5):742–754, 2010.
56. Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 869–877, 2018.
57. D. Flam-Shepherd, T. Wu, P. Friederich, and A. Aspuru-Guzik. Neural message passing on high order paths. *arXiv preprint*, 2020.
58. Rafal Karczewski, Amauri H. Souza, and Vikas Garg. On the generalization of equivariant graph neural networks. In *ICML*, 2024.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

59. Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1365–1374, 2015.
60. Nicolás Serrano, Francisco Castro, and Alfons Juan. The rodrigo database. In *LREC*, pages 19–21, 2010.
61. Jinhyeok Choi, Heehyeon Kim, Minhyeong An, and Joyce Jiyoung Whang. Spot-mamba: Learning long-range dependency on spatio-temporal graphs with selective state spaces. *arXiv preprint arXiv:2406.11244*, 2024.
62. Chanyoung Chung, Jaejun Lee, and Joyce Jiyoung Whang. Representation learning on hyper-relational and numeric knowledge graphs with transformers. In *KDD*, 2023.
63. T. Maehara and H. NT. A simple proof of the universality of invariant/equivariant graph neural networks. *arXiv preprint*, 2019.
64. K. M. Borgwardt. *Graph kernels*. Phd thesis, Ludwig Maximilians University Munich, 2007.
65. P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613, 1998.
66. Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.
67. Kenji Kawaguchi, Zhun Deng, Kyle Luh, and Jiaoyang Huang. Robustness implies generalization via data-dependent generalization bounds. In *ICML*, 2022.
68. V. N. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Avtomatika i Telemekhanika*, 24(6): 937–945, 1964.
69. Judea Pearl. Causality 2002-2020 - introduction. In Hector Geffner, Rina Dechter, and Joseph Y. Halpern, editors, *Probabilistic and Causal Inference: The Works of Judea Pearl*, volume 36 of *ACM Books*, pages 393–398. ACM, 2022.
70. Arnaud Casteigts, Paola Flocchini, Emmanuel Godard, Nicola Santoro, and Masafumi Yamashita. On the expressivity of time-varying graphs. *Theoretical Computer Science*, 590:27–37, 2015.
71. S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. 2014.
72. Trang Pham, Truyen Tran, Dinh Q. Phung, and Svetha Venkatesh. Column networks for collective classification. In *Proc. of AAAI*, pages 2485–2491, 2017.
73. Wolfgang Maass. Bounds for the computational power and learning complexity of analog neural nets. *SIAM Journal on Computing*, 26(3):708–732, 1997.
74. J. Kazius, R. McGuire, and R. Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal Medicinal Chemistry*, (13):312–320, 2005.
75. Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
76. Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):4358–4370, 2024.
77. Zeyu Zhang, Lu Li, Shuyan Wan, Sijie Wang, Zhiyi Wang, Zhiyuan Lu, Dong Hao, and Wanli Li. Dropedge not foolproof: Effective augmentation method for signed graph neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
78. D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.
79. Sami Abu-El-Haija, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. N-gcn: Multi-scale graph convolution for semi-supervised node classification. In *uncertainty in artificial intelligence*, pages 841–851. PMLR, 2020.
80. Nesreen K. Ahmed, Theodore L. Willke, and Ryan A. Rossi. Estimation of local subgraph counts. In *2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, December 5-8, 2016*, pages 586–595, 2016.
81. Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *ICLR*, 2020.
82. Siddhartha Mishra and T Konstantin Rusch. Enhancing accuracy of deep learning algorithms by training with low-discrepancy sequences. *SIAM Journal on Numerical Analysis*, 59(3):1811–1834, 2021.
83. Gauthier Guinet, Behrooz Omidvar-Tehrani, Anoop Deoras, and Laurent Callot. Automated evaluation of retrieval-augmented language models with task-specific exam generation. 2024.
84. Joan Andreu Sánchez, Verónica Romero, Alejandro H Toselli, Mauricio Villegas, and Enrique Vidal. A set of benchmarks for handwritten text recognition on historical documents. *Pattern Recognition*, 94:122–134, 2019.
85. M. Grohe and M. Otto. Pebble games and linear equations. *Journal of Symbolic Logic*, 80(3):797–844, 2015.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2025 doi:10.20944/preprints202507.1774.v1

86. Denis Lukovnikov and Asja Fischer. Improving breadth-wise backpropagation in graph neural networks helps learning long-range dependencies. In *International Conference on Machine Learning*, pages 7180–7191. PMLR, 2021.
87. Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.
88. Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.
89. Brandon M. Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *NeurIPS 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14510–14519, 2019.
90. A. Cardon and M. Crochemore. Partitioning a graph in $O(|A| \log_2 |V|)$. *Theoretical Computer Science*, (1):85 – 98, 1982.
91. Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. Extensions of marginalized graph kernels. In *ICLR*, 2004.
92. Floris Geerts and Juan L Reutter. Expressiveness and approximation properties of graph neural networks. *arXiv preprint arXiv:2204.04661*, 2022.
93. N. Shervashidze, S. V. N. Vishwanathan, T. H. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, pages 488–495, 2009.
94. R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 1987.
95. Liangyue Li, Hanghang Tong, Yanghua Xiao, and Wei Fan. *Cheetah*: Fast graph kernel tracking on dynamic graphs. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 280–288, 2015.
96. Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael M. Bronstein. Bundle neural network for message diffusion on graphs. In *The Thirteenth International Conference on Learning Representations*, 2025.
97. Atsushi Suzuki, Atsushi Nitanda, Taiji Suzuki, Jing Wang, Feng Tian, and Kenji Yamanishi. Tight and fast generalization error bound of graph embedding in metric space. In *ICML*, 2023.
98. Jiawei Zhang and Lin Meng. Gresnet: Graph residual network for reviving deep gnns from suspended animation. *arXiv preprint arXiv:1909.05729*, 2019.
99. Barbara Hammer. Generalization ability of folding networks. *IEEE Trans. Knowl. Data Eng.*, (2):196–206, 2001.
100. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2013.
101. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *ACM Symposium on Computational Geometry*, pages 253–262, 2004.
102. Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. Training graph neural networks with 1000 layers. In *ICML*, 2021.
103. Kenta Oono and Taiji Suzuki. Optimization and generalization analysis of transduction through gradient boosting and application to multi-scale graph neural networks. In *NeurIPS*, 2020.
104. R. Levie, W. Huang, L. Bucci, M. Bronstein, and G. Kutyniok. Transferability of spectral graph convolutional neural networks. *Journal of Machine Learning Research*, 22(272):1–59, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.