

A Comprehensive Literature Review with Self-Reflection

Literature Review

October 7, 2025

Abstract

This literature review provides a comprehensive analysis of recent research in the field. The review synthesizes findings from 328 research papers, identifying key themes, methodological approaches, and future research directions.

Contents

References

121

1. Introduction

The landscape of machine learning has undergone a profound transformation with the advent of Graph Neural Networks (GNNs), a paradigm specifically engineered to process and learn from data structured as graphs (121; 123; 216). Unlike traditional machine learning models that primarily operate on Euclidean data (e.g., images, text sequences, tabular data), GNNs are uniquely positioned to leverage the rich relational information and intricate topological structures inherent in graph-structured data (164). This capability is paramount in an increasingly interconnected world, where complex systems across science, technology, and society are naturally represented as graphs. From social networks and biological systems to molecular structures, transportation networks, and knowledge graphs, the ubiquity of graph-structured data has rendered GNNs an indispensable tool for extracting meaningful insights and making accurate predictions (42; 235).

The fundamental motivation behind the development of GNNs stems from the inherent limitations of conventional deep learning architectures when confronted with non-Euclidean data. Convolutional Neural Networks (CNNs), for instance, excel at processing grid-like data by exploiting local connectivity and translational invariance, but they lack a natural mechanism to handle the irregular and often dynamic connectivity patterns of graphs (121). Similarly, Recurrent Neural Networks (RNNs) and Transformers are designed for sequential data, struggling to capture the non-linear, multi-directional dependencies prevalent in graphs. GNNs address this critical gap by extending the principles of deep learning, particularly the concept of localized feature aggregation, to arbitrary graph topologies. They operate on the premise of "message passing," where nodes iteratively aggregate information from their neighbors, thereby learning representations that encode both local structural patterns and node features (?). This iterative message passing mechanism allows GNNs to effectively capture relational inductive biases, enabling them to model complex interactions and dependencies that are crucial for understanding graph-structured phenomena (?).

The growing importance of GNNs is evidenced by their widespread adoption and state-of-the-art performance across an astonishingly diverse array of fields. In recom-

mender systems, GNNs have revolutionized personalized recommendations by modeling user-item interactions as graphs, capturing high-order relationships that significantly enhance prediction accuracy (4; 111; 89; 79; 35; 92; 155; 137; 127; 184; 136; 153; 182; 223; 234; 259; 171; 320). In drug discovery and materials science, they are instrumental in predicting molecular properties and interactions by treating molecules as graphs, accelerating the design of new compounds and materials (165; 51; 124; 99; 175; 188; 169; 277; 271; 275; 289; 287; 298; 253; 254; 244; 274). Computer vision has seen GNNs applied to tasks ranging from object detection and scene understanding to point cloud processing and human pose estimation, where they model relationships between visual entities or geometric components (7; 125; 134; 176; 193; 266; 267). Furthermore, GNNs are making significant inroads into critical infrastructure domains such as intelligent transportation systems (130; 110; 170; 263), wireless communication networks (149; 154; 204; 291), and power systems (172; 58; 276), optimizing resource allocation and predicting system dynamics. Their utility extends to cybersecurity for intrusion detection and anomaly detection (21; 191; 186; 180; 142; 221; 225; 227; 273; 321) and even to epidemic modeling for forecasting disease spread (196; 69).

Despite their remarkable success, the development of GNNs has not been without its challenges. Early GNNs, often based on spectral graph theory or simplified message-passing schemes, quickly encountered limitations in expressiveness, struggling to distinguish between non-isomorphic graphs that are structurally distinct (122; 65; 133; 55; 60; 160; 189; 53; 101; 314; 251; 301; 299; 319). A pervasive issue, particularly in deep GNNs, is "over-smoothing," where node representations converge to indistinguishable values across layers, leading to a loss of discriminative power (52; 115; 81; 32; 59; 18; 230; 282). Furthermore, many foundational GNNs implicitly assume "homophily," meaning connected nodes share similar features or labels. This assumption breaks down in "heterophilous" graphs, common in real-world scenarios, where connected nodes are often dissimilar (72; 128; 217; 113; 108; 208; 211; 103). Scalability to large graphs, robustness against adversarial attacks (26; 87; 104; 74; 82; 83; 36; 109; 262; 317; 248; 250; 306), and the critical need for interpretability and fairness (19; 31; 105; 9; 11; 107; 43; 39; 64; 54; 94; 66;

61; 76; 38; 28; 212; 67; 264; 322) represent ongoing research frontiers. The field has also recognized the necessity for more rigorous and realistic evaluation methodologies, moving beyond superficial comparisons to truly assess model capabilities (5; 44).

This comprehensive literature review aims to consolidate the vast and rapidly evolving body of knowledge surrounding GNNs. It seeks to provide a structured understanding of their foundational principles, evolutionary trajectory, diverse methodologies, and far-reaching applications. By critically analyzing the strengths and limitations of various GNN architectures and techniques, this review endeavors to identify persistent research gaps and illuminate promising future directions. The subsequent sections will delve into the evolution of GNN architectures, explore advanced methodologies, survey their applications across various domains, and discuss the critical challenges and future opportunities that will shape the next generation of graph machine learning.

1.1. The Rise of Graph Neural Networks

The emergence of Graph Neural Networks (GNNs) marks a pivotal moment in machine learning, driven by the increasing recognition that much of the world's data is inherently relational and best represented as graphs (121; 123). Traditional machine learning, particularly deep learning, has achieved monumental success on data with regular grid structures like images (CNNs) and sequences like text (RNNs, Transformers). However, these architectures fundamentally struggle with the irregular, non-Euclidean nature of graphs, where nodes have varying numbers of neighbors and no inherent ordering. This fundamental mismatch motivated the development of GNNs, which generalize the concept of convolution and aggregation to arbitrary graph structures (164; 118).

The foundational concept of GNNs can be traced back to early works that attempted to extend neural networks to graphs by iteratively propagating information across edges (? ?). These early models laid the groundwork for the "message passing" paradigm, where each node updates its representation by aggregating information from its immediate neighbors and its own previous state. This iterative process allows nodes to gather increasingly rich contextual information from their local neighborhoods, effectively learn-

ing representations that capture both node features and structural roles. The resurgence of interest in GNNs was significantly catalyzed by the introduction of models like Graph Convolutional Networks (GCNs) (1) and Graph Attention Networks (GATs) (2), which provided scalable and effective implementations of this message-passing framework. GCNs, for instance, simplified spectral graph convolutions into a first-order approximation, demonstrating compelling performance on semi-supervised node classification tasks (3). Building on this, GATs introduced an attention mechanism, allowing nodes to assign different weights to their neighbors, thereby learning more flexible and powerful aggregation functions (4). This innovation addressed a key limitation of GCNs, which treated all neighbors equally, and exemplified the field's shift towards more adaptive and expressive aggregation schemes.

The initial success of these models quickly highlighted both the immense potential and the inherent limitations of the nascent GNN paradigm. A critical tension emerged regarding the "expressive power" of GNNs, specifically their ability to distinguish between non-isomorphic graphs. Early theoretical analyses, notably by (122) and (65), demonstrated that many standard GNNs are no more powerful than the 1-dimensional Weisfeiler-Lehman (1-WL) test, a heuristic for graph isomorphism. This implies that these GNNs cannot differentiate between certain structurally distinct graphs, limiting their capacity to capture complex graph patterns (133; 55). This limitation spurred significant research into developing more expressive GNN architectures, such as higher-order GNNs (65) or those incorporating richer structural information. For example, (6) introduced Path Neural Networks (PathNNs) which explicitly leverage path information, showing that by operating on "annotated sets of paths," their variants can surpass the expressive power of the 1-WL algorithm and even distinguish graphs indistinguishable by the 3-WL algorithm. This directly addresses the expressiveness bottleneck by moving beyond simple local neighborhood aggregation to capture more global and complex structural patterns.

Another significant challenge that quickly became apparent was "over-smoothing," where repeated message passing in deep GNNs causes node representations to become increasingly similar, eventually converging to a single point, thus losing discriminative power

(115; 81; 32; 59). This phenomenon fundamentally limits the depth of GNNs, contrasting sharply with the success of very deep architectures in other domains like computer vision. Various strategies have been proposed to mitigate over-smoothing, including residual connections (52; 18), regularization techniques like DropEdge (?), and more sophisticated aggregation schemes. For instance, (3) introduces FROND, a framework that leverages fractional calculus to generalize continuous GNNs. By using fractional-order derivatives, FROND inherently integrates the entire historical trajectory of node features, enabling memory-dependent dynamics and analytically demonstrating an algebraic (slower) rate of convergence to stationarity, thereby mitigating over-smoothing compared to the exponential convergence of integer-order models. This represents a novel theoretical and architectural approach to a persistent problem.

Furthermore, the implicit assumption of "homophily" (the tendency of connected nodes to be similar) in many early GNNs posed a significant hurdle for real-world graphs that often exhibit "heterophily" (connected nodes being dissimilar) (72; 128). Social networks, for example, might connect individuals with diverse interests, while protein-protein interaction networks can link functionally distinct proteins. This tension between model assumption and data reality led to the development of GNNs specifically designed to handle heterophilous graphs. (2) addresses this by proposing GloGNN, which learns a signed coefficient matrix for *all* nodes, effectively capturing global homophily and heterophily simultaneously. Instead of relying on fixed local neighborhoods, GloGNN performs global aggregation with learned coefficients, allowing it to adaptively assign positive coefficients to homophilous nodes and negative/small positive coefficients to heterophilous ones. Crucially, it achieves this with linear time complexity, overcoming the computational burden of naive global aggregation. This exemplifies a critical evolution in GNN design, moving from local, homophily-biased aggregation to more adaptive, global, and heterophily-aware mechanisms. The emergence of non-convolutional GNNs, such as the Random Walk with Unifying Memory (RUM) neural network proposed by (1), further illustrates the field's drive to overcome these fundamental limitations. RUM, by processing graph information solely through random walk trajectories and RNNs, offers a joint remedy to limited expres-

siveness, over-smoothing, and over-squashing, demonstrating superior theoretical properties and competitive empirical performance without relying on traditional convolution operators. This signifies a paradigm shift, exploring alternatives to the message-passing framework to unlock deeper and more robust graph representations.

1.2. Importance of Graph-Structured Data

The pervasive nature of graph-structured data in the modern world underscores the critical importance of Graph Neural Networks (GNNs). Graphs serve as a natural and powerful abstraction for complex systems where entities (nodes) interact or relate to one another (edges) (164; 146). Unlike tabular data or sequences, graphs explicitly encode relational information, which is often the most critical aspect for understanding system behavior and making accurate predictions. This section elaborates on why graph-structured data is so important and how GNNs are uniquely positioned to unlock its value.

One of the primary reasons for the importance of graph-structured data lies in its ability to model intricate relationships that are often overlooked or simplified by other data representations. Consider social networks, where individuals are nodes and friendships or interactions are edges. Understanding influence, community detection, or personalized recommendations fundamentally relies on analyzing these complex relational patterns (79; 155). Traditional machine learning models, if applied to such data, would typically flatten the graph into node features (e.g., demographics) or aggregate edge features, thereby losing the rich topological context. GNNs, through their message-passing mechanism, directly operate on this relational structure, allowing information to flow and aggregate across the graph, thereby learning representations that intrinsically encode a node’s position within the network and its interactions with neighbors (121). This relational inductive bias is what makes GNNs so powerful for graph-structured data.

The impact of GNNs on diverse scientific and industrial domains further highlights the importance of graph data. In biology and chemistry, molecules are inherently graphs, with atoms as nodes and chemical bonds as edges. GNNs have become indispensable for tasks such as predicting molecular properties, drug-target interactions, and protein-

protein interactions (165; 51; 124; 99; 188; 214; 178; 313; 242). For instance, (169) introduces Mole-BERT, rethinking pre-training GNNs for molecules, showcasing the specialized adaptations required to capture intricate chemical information. Similarly, in neuroscience, brain connectivity networks are modeled as graphs, enabling GNNs to analyze brain disorders and understand neural activity patterns (93; 27; 39; 236; 280; 278; 270). The ability of GNNs to learn from these complex, irregular structures has opened new avenues for discovery and understanding in these fields.

Beyond scientific applications, graph data is crucial in various real-world systems. Recommender systems, as extensively surveyed by (4; 89; 92), represent user-item interactions as bipartite graphs, where GNNs can effectively capture high-order collaborative filtering signals, leading to more accurate and diverse recommendations (111; 137; 35). In urban computing, GNNs are employed for traffic flow forecasting, ride-sharing optimization, and intelligent transportation systems, where roads, sensors, and vehicles form dynamic graphs (130; 126; 110; 170). For example, (30) provides a comprehensive survey on GNNs for time series, covering forecasting, classification, imputation, and anomaly detection, underscoring their versatility in dynamic graph settings. The financial sector benefits from GNNs in fraud detection, where transactions and entities form complex networks, allowing GNNs to identify anomalous patterns indicative of fraudulent activities (? 246; 297; 312). Even in computer vision, GNNs are increasingly used to model relationships between objects in a scene, parse human poses, or process 3D point clouds and meshes, as detailed in the task-oriented survey by (7).

The importance of graph-structured data also brings forth unique challenges that GNNs are designed to address. The inherent non-Euclidean nature means that standard operations like convolution or pooling, which assume fixed-size inputs and local translational invariance, are not directly applicable. GNNs overcome this by defining aggregation functions that are permutation-invariant to the order of neighbors, ensuring that the learned representations are independent of how the adjacency list is ordered (122). Furthermore, real-world graphs are often dynamic, evolving over time with new nodes and edges appearing or disappearing. This necessitates the development of dynamic GNNs

capable of adapting to continuous changes, as highlighted by works like (33) on temporal graphs and (228) on spatio-temporal distribution shifts. The challenge of heterophily, where connected nodes are dissimilar, is another critical aspect of real-world graph data. While early GNNs struggled with this, models like GloGNN (2) demonstrate how to effectively capture global homophily even in heterophilous settings by learning adaptive, signed aggregation coefficients, showcasing the field’s progress in handling complex graph properties. The need for robust and fair GNNs is also paramount given the sensitivity of graph data in applications like social networks and healthcare (9; 11; 98; 116; 12; 25; 88; 322). This continuous innovation in GNN architectures and training methodologies underscores the profound and growing importance of graph-structured data as a fundamental data type in modern machine learning.

1.3. Scope and Structure of the Review

This comprehensive literature review is meticulously designed to provide a structured and critical understanding of Graph Neural Networks (GNNs), spanning their foundational principles, evolutionary trajectory, diverse methodologies, and far-reaching applications. Given the rapid proliferation of research in this domain, a systematic consolidation of knowledge is essential to identify key advancements, persistent challenges, and promising future directions. The scope of this review encompasses the core architectural designs, theoretical underpinnings, practical considerations, and societal implications of GNNs, while also highlighting critical evaluation practices.

The review begins by establishing the fundamental motivation for GNNs, recognizing the inherent limitations of traditional machine learning models when confronted with non-Euclidean, graph-structured data. As discussed in the preceding subsections, the ubiquity of relational information across scientific, technological, and social domains necessitates specialized tools capable of effectively processing and learning from graphs (121; 123; 164). This introductory section sets the stage by briefly touching upon the transformative impact GNNs have achieved and the critical challenges they address, guiding the reader through the subsequent, more detailed exploration.

The subsequent sections of this review will be structured to provide a logical and progressive understanding of the GNN landscape. We will first delve into the **Evolution of GNN Architectures**, tracing their development from early spectral and spatial approaches to more advanced and specialized designs. This section will critically compare foundational models like GCNs (?) and GATs (?) with later innovations that address limitations such as expressiveness (122; 6), over-smoothing (115; 3), and scalability (201; 198; 168). For instance, the shift from purely convolutional designs to non-convolutional, walk-based architectures like RUM (1) will be analyzed as a response to fundamental architectural bottlenecks.

Following this, the review will explore **Advanced Methodologies and Training Paradigms** in GNNs. This includes a detailed examination of techniques for handling graph heterogeneity (2; 108; 208), dynamic graphs (33; 110), and large-scale graphs (201). We will also cover advancements in self-supervised learning (20; 56; 148), pre-training strategies (49; 112; 24; 15; 77; 166), and data augmentation techniques (13). A critical analysis of the trade-offs between model complexity, computational efficiency, and performance will be central to this section.

The review will then transition to a comprehensive overview of **Applications of GNNs Across Diverse Domains**. This section will systematically categorize and discuss the impact of GNNs in areas such as recommender systems (4; 89), computer vision (7), natural language processing (162; 41; 239; 73), bioinformatics and materials science (51; 165; 167), and critical infrastructure management (154; 172; 170). Each application area will be accompanied by examples of specific problems GNNs solve, their core innovations in that context, and their demonstrated empirical success.

Finally, the review will dedicate a section to **Critical Challenges, Trustworthiness, and Future Directions**. This crucial part will synthesize the unresolved debates and persistent limitations within the GNN field, including issues of interpretability and explainability (8; 105; 43; 76), fairness and bias mitigation (9; 11; 98; 12), and robustness against adversarial attacks (26; 87; 104; 82; 36). A particular focus will be placed on the need for rigorous and realistic evaluation methodologies, as highlighted by (5), which

critically examines current pitfalls in benchmarking link prediction models. This section will also project future research trajectories, such as the integration of GNNs with large language models (287), quantum computing (324), and novel theoretical frameworks like fractional calculus (3).

By adopting this structured approach, this literature review aims to consolidate fragmented knowledge, identify critical research gaps, and provide a holistic, analytical understanding of the GNN landscape. It serves as a valuable resource for both seasoned researchers and newcomers, guiding them through the complexities and exciting opportunities in this rapidly evolving field.

2. Foundational Concepts and Early Architectures

The emergence of Graph Neural Networks (GNNs) represents a paradigm shift in machine learning, offering a powerful framework to process and learn from data intrinsically structured as graphs. This section delves into the foundational concepts that underpin GNNs, tracing their intellectual lineage from the essential background of graph representation learning to the development of early, influential architectural designs. At its core, GNNs leverage the "message-passing" paradigm, an iterative process where nodes aggregate information from their local neighborhoods to update their representations. This mechanism, while intuitive and effective for many tasks, also introduces inherent limitations, particularly concerning the models' discriminative power. A critical aspect explored here is the theoretical expressiveness of GNNs, benchmarked against the Weisfeiler-Leman (WL) graph isomorphism test, which provides a formal measure of their ability to distinguish between structurally different graphs. By establishing the initial capabilities, core design principles, and inherent limitations of these early GNN models, this section lays the groundwork for understanding the subsequent advancements and the ongoing research efforts to overcome these fundamental challenges.

The journey into GNNs begins with the fundamental problem of graph representation learning: how to embed complex, irregular graph structures into low-dimensional vector spaces such that the topological and feature information is preserved and useful

for downstream tasks (121; 123). Unlike Euclidean data, graphs lack a canonical ordering of nodes or a fixed neighborhood size, making direct application of traditional deep learning architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), challenging. Early approaches to graph learning often relied on handcrafted features or matrix factorization techniques, which struggled with scalability and the ability to capture complex, non-linear patterns. GNNs address this by extending the principles of deep learning to graphs, allowing models to learn features directly from the graph structure and node attributes. This capability is crucial because many real-world systems, from social networks and biological pathways to knowledge graphs and transportation systems, are naturally represented as graphs, where relationships between entities are as important as the entities themselves (164; 118).

The core innovation of early GNNs lies in their ability to iteratively aggregate and transform information across the graph. This iterative process, often referred to as message passing, enables nodes to gather increasingly rich contextual information from their multi-hop neighborhoods. Each layer of a GNN can be conceptualized as a step in this information propagation, where a node's representation is updated based on its previous state and the aggregated messages from its neighbors. This local aggregation mechanism, inspired by classical signal processing on graphs, forms the backbone of most GNN architectures. However, the simplicity and local nature of this paradigm also lead to identifiable limitations. For instance, the implicit assumption of "homophily," where connected nodes tend to be similar, often restricts the performance of many early GNNs on "heterophilous" graphs, where connected nodes are frequently dissimilar (72; 128). Furthermore, the repeated aggregation can lead to "over-smoothing," a phenomenon where node representations become indistinguishable as the number of layers increases, thereby limiting the depth and discriminative power of GNNs (115; 81).

A critical theoretical lens through which the capabilities of early GNNs are evaluated is their relationship to the Weisfeiler-Leman (WL) graph isomorphism test (?). This test provides a hierarchical framework for assessing the discriminative power of graph algorithms, with the 1-WL test serving as a common benchmark for many message-passing

GNNs. Pioneering work by (122) formally established that many popular GNNs are at most as powerful as the 1-WL test, meaning they cannot distinguish between certain non-isomorphic graphs that the 1-WL test fails to differentiate. This theoretical limitation spurred significant research into designing more expressive GNN architectures, such as higher-order GNNs (65) or those that incorporate richer structural information like paths (6). The tension between achieving high expressiveness, computational efficiency, and scalability remains a central theme in GNN research, with early models often sacrificing one for the others. This section will systematically unpack these foundational elements, highlighting the initial successes, inherent trade-offs, and the critical questions that continue to drive innovation in the field.

Graph Representation Learning Fundamentals

Graph representation learning forms the bedrock upon which Graph Neural Networks (GNNs) are built, addressing the fundamental challenge of transforming complex, non-Euclidean graph structures into a format amenable to machine learning algorithms (121; 123). The core objective is to embed nodes, edges, or entire graphs into low-dimensional vector spaces, often referred to as "embeddings," such that their structural roles, features, and relational properties are preserved. These embeddings can then be used for various downstream tasks, including node classification, link prediction, and graph classification (164; 118).

Historically, graph analysis relied on handcrafted features (e.g., node degrees, clustering coefficients, PageRank scores) or matrix factorization techniques applied to adjacency or Laplacian matrices. While these methods provided some insights, they suffered from several limitations. Handcrafted features often required domain expertise, were not generalizable across different graph types, and struggled to capture complex, multi-hop dependencies. Matrix factorization methods, such as those used in early network embedding techniques like DeepWalk (?) and Node2Vec (?), could learn embeddings by modeling random walks or neighborhood similarities. However, these methods were typically transductive, meaning they could not generalize to unseen nodes or graphs without retraining,

and often decoupled feature learning from the end task, limiting their effectiveness.

The advent of GNNs marked a significant departure, introducing an end-to-end, inductive learning paradigm for graph representations. The fundamental idea is to learn a mapping function that transforms a node’s features and its local graph structure into a dense vector representation. This is achieved through an iterative process of feature aggregation and transformation. For a given node v , its embedding at layer k , denoted as $\mathbf{h}_v^{(k)}$, is typically computed by aggregating information from its neighbors and its own representation from the previous layer $k - 1$. This process can be generalized as: $\mathbf{h}_v^{(k)} = \text{UPDATE}^{(k)}(\mathbf{h}_v^{(k-1)}, \text{AGGREGATE}^{(k)}(\{\mathbf{h}_u^{(k-1)} \mid u \in \mathcal{N}(v)\}))$, where $\mathcal{N}(v)$ denotes the set of neighbors of node v , and AGGREGATE and UPDATE are learnable functions, often implemented as neural networks (?).

This framework inherently addresses the non-Euclidean nature of graphs by operating directly on the graph topology. The permutation invariance of the aggregation function (e.g., sum, mean, max) ensures that the learned representations are independent of the order in which neighbors are processed, a crucial property for graph data. This mechanism allows GNNs to capture relational inductive biases, meaning they learn patterns that are robust to permutations of node indices and can generalize to graphs with different structures. For instance, a GNN can learn that a node with many high-degree neighbors plays a specific structural role, regardless of the specific identities of those neighbors.

However, this foundational approach also introduced several critical challenges. A primary concern is the **homophily assumption** (72). Many early GNNs, by design, implicitly assume that connected nodes share similar features or labels. This is because the aggregation functions typically average or sum neighbor features, effectively smoothing them. While effective for homophilous graphs (e.g., citation networks where connected papers are often on similar topics), this assumption severely limits performance on **heterophilous graphs** (e.g., protein-protein interaction networks where connected proteins might have diverse functions) (128; 217). The field recognized this tension, leading to the development of models like GloGNN (2), which explicitly addresses heterophily by learning a *signed coefficient matrix* for global aggregation, allowing it to assign positive

weights to homophilous nodes and negative or small positive weights to heterophilous ones. This innovation demonstrates a critical evolution from simple local smoothing to adaptive, global, and heterophily-aware aggregation.

Another significant challenge in graph representation learning, particularly as GNNs become deeper, is **over-smoothing** (115; 81). As nodes repeatedly aggregate information from their neighbors across multiple layers, their representations tend to converge to a single, indistinguishable vector, losing their unique discriminative power. This phenomenon fundamentally limits the effective depth of GNNs, contrasting with the success of very deep architectures in other domains. While techniques like residual connections (52; 18) and dropout variants (?) were introduced to mitigate this, the problem persists. A novel approach from (3) introduces the FRactional-Order graph Neural Dynamical network (FROND), which generalizes continuous GNNs by employing Caputo fractional derivatives. This allows FROND to inherently integrate memory-dependent dynamics into node updates, leading to an algebraic (slower) rate of convergence to stationarity, thereby analytically mitigating over-smoothing compared to the exponential convergence of integer-order models. This highlights a critical shift towards understanding and leveraging the temporal dynamics of information propagation to build more robust representations.

The efficiency and scalability of graph representation learning are also paramount. Early GNNs often required access to the entire graph during training, which is infeasible for large-scale graphs. This led to the development of sampling-based approaches like GraphSAGE (?), which samples a fixed number of neighbors for aggregation, making training mini-batch compatible and scalable. The challenge of evaluating these learned representations also became apparent. As highlighted by (5), current evaluation practices for tasks like link prediction often suffer from pitfalls such as inconsistent data splits, poor hyperparameter tuning, and unrealistic negative sampling. Their proposed Heuristic Related Sampling Technique (HeaRT) addresses this by generating *harder*, more realistic negative samples, providing a more robust benchmark for assessing the true capabilities of GNNs. This emphasizes that the quality of learned representations is not only dependent on the model architecture but also on the rigor of its evaluation. In essence, graph

representation learning provides the necessary foundation for GNNs to operate, but its inherent challenges have continuously pushed the field towards more sophisticated and robust architectural designs, aggregation mechanisms, and evaluation methodologies.

The Message Passing Paradigm and Early Models

The message-passing paradigm is the cornerstone of most Graph Neural Networks (GNNs), providing an elegant and unified framework for information propagation and aggregation across graph structures (Wu et al., 2021). This paradigm formalizes the intuitive idea that a node's representation can be iteratively refined by exchanging "messages" with its neighbors and combining these messages with its own current state. This section elucidates the core mechanics of message passing and introduces early, influential GNN architectures that instantiated this principle, critically examining their innovations and inherent limitations.

The general message-passing framework, as formalized by (Wu et al., 2021), involves two main steps at each layer k :

Message Generation: For each node v , messages $m_{uv}^{(k)}$ are generated from its neighbors $u \in \mathcal{N}(v)$, often as a function of their previous layer's embeddings $\mathbf{h}_u^{(k-1)}$ and the edge features e_{uv} .

Aggregation and Update: Each node v aggregates the incoming messages from its neighbors, $\sum_{u \in \mathcal{N}(v)} m_{uv}^{(k)}$, and then updates its own embedding $\mathbf{h}_v^{(k)}$ by combining this aggregated message with its previous embedding $\mathbf{h}_v^{(k-1)}$.

This iterative process allows information to flow across the graph, enabling nodes to capture increasingly global structural and feature information as the number of layers increases. The permutation invariance of the aggregation function (e.g., sum, mean, max) is crucial, ensuring that the node embeddings are independent of the arbitrary ordering of neighbors.

One of the earliest and most influential instantiations of this paradigm was the **Graph Convolutional Network (GCN)**, proposed by (Kipf & Wainwright, 2016). * **Context:** GCNs aimed to generalize the concept of convolution from grid-like data (images) to arbitrary graph structures, drawing inspiration from spectral graph theory but simplifying it for practical application.

They sought to address the challenge of semi-supervised node classification efficiently. *

Mechanism: GCNs achieve message passing through a simplified, first-order approximation of spectral graph convolutions. The core operation for updating node features $\mathbf{H}^{(k)}$ at layer k is given by $\mathbf{H}^{(k)} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(k-1)} \mathbf{W}^{(k)}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\tilde{\mathbf{D}}$ is its degree matrix, and $\mathbf{W}^{(k)}$ is a learnable weight matrix. This effectively averages the features of a node’s neighbors (including itself) and applies a linear transformation. * **Conditions for Success:** GCNs demonstrated strong performance on semi-supervised node classification tasks, particularly on homophilous graphs like citation networks (e.g., Cora, Citeseer, Pubmed) (?). Their simplicity and efficiency made them a popular baseline. * **Theoretical Limitations:** Despite their success, GCNs are theoretically limited in their expressive power, being at most as powerful as the 1-Weisfeiler-Leman (1-WL) test (122). This means they cannot distinguish between certain non-isomorphic graphs, limiting their ability to capture complex structural patterns. * **Practical Limitations:** GCNs suffer from **over-smoothing** when stacked deeply (115; 81), where node representations become indistinguishable. They also employ fixed, unweighted aggregation, implicitly assuming all neighbors contribute equally, which is suboptimal for graphs with varying neighbor importance or heterophily (72).

Building on the message-passing framework and addressing some limitations of GCNs, **Graph Attention Networks (GATs)** were introduced by (?). * **Context:** GATs aimed to overcome the fixed-weight aggregation of GCNs by allowing the model to learn adaptive weights for each neighbor, making the aggregation process more flexible and powerful. * **Mechanism:** GATs introduce an attention mechanism where each node computes an attention coefficient for each of its neighbors, indicating the importance of that neighbor’s features. These coefficients are then used to compute a weighted sum of neighbor features. The attention mechanism is typically implemented using a single-layer feedforward neural network and a LeakyReLU activation, followed by a softmax normalization over neighbors. Multiple attention heads can be used to stabilize the learning process. * **Conditions for Success:** GATs showed improved performance over GCNs on various node classification and graph classification tasks, especially in scenarios where

different neighbors have varying importance or when dealing with inductive settings (?). * **Theoretical Limitations:** Despite their adaptive aggregation, GATs are still generally considered to be at most as powerful as the 1-WL test (122), inheriting the same expressiveness limitations as GCNs. * **Practical Limitations:** GATs still face the over-smoothing problem in deep architectures (230). The computational cost of computing attention coefficients for every neighbor can also be higher than GCNs, especially for dense graphs.

The limitations of these early message-passing models, particularly over-smoothing and restricted expressiveness, spurred significant research into developing deeper and more robust GNNs. For instance, **deep GNNs** attempted to mitigate over-smoothing through architectural innovations like residual connections, similar to those in CNNs (52; 18). However, these often only partially alleviated the problem, as the fundamental smoothing operation remained. A more radical departure is seen in the **FRactional-Order graph Neural Dynamical network (FROND)** framework (3). * **Context:** FROND addresses the limitations of traditional continuous GNNs, which rely on integer-order differential equations and struggle to capture long-term dependencies and memory effects, leading to over-smoothing. ***Mechanism:** It replaces the integer-order differential operator with the Caputo fractional derivative. This allows FROND to inherently integrate the entire historical trajectory of node features, enabling memory-dependent dynamics. The paper analytically shows that this leads to a slow algebraic rate of convergence to stationarity, thereby mitigating over-smoothing compared to the exponential* convergence of Markovian integer-order models.* * **Comparison:** FROND represents a generalization of existing continuous GNNs, offering a novel theoretical foundation for building deeper and more robust models by leveraging non-local, memory-dependent dynamics. * **Implications:** This approach offers a new class of GNNs that can model complex, non-Markovian feature updates, potentially unlocking deeper insights into graph dynamics and providing a principled way to overcome over-smoothing.

Another significant innovation challenging the conventional message-passing paradigm is the **Random Walk with Unifying Memory (RUM) neural network** (1). *Con-*

text: RUM directly confronts the fundamental technical problems of convolution-based* GNNs: limited expressiveness (beyond 1-WL), over-smoothing, and over-squashing (difficulty learning long-range dependencies). ***Mechanism:*** RUM is an entirely non-convolutional* architecture. Instead of aggregating messages from direct neighbors, it samples finite-length random walks for each node. An RNN (specifically GRU) processes the sequence of node embeddings along these walks, merging "semantic representations" (node features) with "anonymous experiment" features (topological environment based on unique node occurrences in the walk). Node representations are then formed by averaging the unifying memories of walks terminating at that node. * ***Innovation:*** The "anonymous experiment" is a novel way to encode topological context, and the use of RNNs on random walks provides a fundamentally different information propagation mechanism. * ***Theoretical Contributions:*** RUM is theoretically shown to be more expressive than the 1-WL test, capable of distinguishing graphs that 1-WL equivalent GNNs cannot. It also theoretically and empirically attenuates over-smoothing (non-diminishing Dirichlet energy) and alleviates over-squashing (slower decay in inter-node Jacobian). * ***Comparison:*** RUM offers a compelling alternative to the message-passing paradigm, jointly addressing multiple long-standing GNN limitations without relying on convolution operators. Its runtime complexity is agnostic to the number of edges, making it highly scalable. * ***Implications:*** RUM signifies a potential paradigm shift, suggesting that effective graph learning can occur outside the traditional message-passing framework, opening avenues for more robust and scalable GNN designs.

In summary, the message-passing paradigm provided a powerful initial framework for GNNs, leading to influential models like GCN and GAT. However, their inherent limitations in expressiveness, over-smoothing, and handling heterophily quickly became apparent. These challenges have driven the field to explore more sophisticated aggregation mechanisms, novel theoretical foundations like fractional calculus, and entirely new non-convolutional architectures, pushing the boundaries of what GNNs can achieve.

Theoretical Expressiveness: The Weisfeiler-Leman Test

The theoretical expressiveness of Graph Neural Networks (GNNs) is a critical dimension for understanding their capabilities and limitations, particularly their power to distinguish between non-isomorphic graphs. This aspect is predominantly benchmarked against the **Weisfeiler-Leman (WL) graph isomorphism test**, a hierarchy of algorithms designed to test graph isomorphism (?). This section delves into the relationship between GNNs and the WL test, highlighting how this theoretical framework has shaped the development of more powerful GNN architectures.

Graph isomorphism is the problem of determining whether two graphs are structurally identical, even if their nodes are labeled differently. This is a fundamental problem in graph theory, and its computational complexity remains an open question. For GNNs, the ability to distinguish non-isomorphic graphs is crucial because it dictates their capacity to learn unique representations for structurally distinct inputs, which is essential for tasks like graph classification or property prediction. If a GNN cannot differentiate between two non-isomorphic graphs, it will assign them identical embeddings, leading to incorrect predictions for tasks that depend on structural nuances.

The **1-Weisfeiler-Leman (1-WL) test**, also known as the color refinement algorithm, serves as the most common benchmark for GNN expressiveness. * **Mechanism:** The 1-WL test iteratively refines node "colors" (labels) based on the multiset of colors of their neighbors. Initially, all nodes are assigned a color based on their features (or a default color if no features exist). In each iteration, a node's new color is determined by hashing its current color together with the multiset of colors of its neighbors. This process continues until no node's color changes, or a fixed number of iterations is reached. If two graphs can be distinguished by their final color distributions, they are non-isomorphic. However, if they have the same final color distribution, the 1-WL test cannot distinguish them, even if they are non-isomorphic. ***Connection to GNNs:** Pioneering work by (122) and (65) formally established a deep connection between the message-passing paradigm of GNNs and the 1-WL test. They demonstrated that many standard message-passing GNNs, including GCNs (?) and GATs (?), are at most as powerful as the 1-WL test*. This*

means that if the 1-WL test cannot distinguish two non-isomorphic graphs, these GNNs also cannot distinguish them. This theoretical insight provided a crucial upper bound on the discriminative power of a broad class of GNNs. * **Limitations of 1-WL:** The 1-WL test, and consequently 1-WL equivalent GNNs, fail to distinguish many common non-isomorphic graphs. For example, they cannot differentiate between regular graphs (where all nodes have the same degree) that are non-isomorphic, or certain pairs of graphs with the same degree sequence but different higher-order structural properties (e.g., specific cycle counts) (60; 133). This limitation implies that standard GNNs might struggle with tasks requiring fine-grained structural understanding, such as counting substructures (251).

The recognition of this expressiveness bottleneck sparked a significant research direction: designing GNNs that surpass the 1-WL test. * **Higher-Order GNNs:** (65) proposed higher-order GNNs, such as the k -WL GNN, which leverage k -tuples of nodes (instead of individual nodes) to perform message passing, effectively mimicking higher-dimensional WL tests. While theoretically more expressive (e.g., 2-WL can distinguish all regular graphs), these models often incur significantly higher computational complexity, scaling polynomially with k and the number of nodes, making them less practical for large graphs. * **Path Neural Networks (PathNNs)** (6) offer a compelling alternative to enhance expressiveness by explicitly leveraging path information. * **Context:** PathNNs directly address the 1-WL limitation by recognizing that paths encode richer structural information than local neighborhoods alone. * **Mechanism:** PathNNs encode paths of various lengths emanating from each node using recurrent layers. The key innovation lies in operating on "annotated sets of paths," where nodes within paths are recursively labeled with hashes of their shorter annotated path sets. This hierarchical encoding enriches the structural context beyond simple neighborhood aggregation. * **Theoretical Evidence:** (6) formally demonstrated that their **AP-Trees** (All Simple Paths) are strictly more powerful than **WL-Trees**, and crucially, that PathNN variants operating on annotated sets of paths ($\sim\text{SP}$, $\sim\text{SP}+$, $\sim\text{AP}$) are strictly more powerful than the 1-WL algorithm. The most expressive variant ($\sim\text{AP}$) was even shown to distinguish graphs that are 3-WL

indistinguishable. * **Limitations:** While powerful, finding all simple paths (especially for the AP variant) is NP-hard, necessitating approximations by fixing a maximum path length K . The computational complexity can still be a practical limitation for dense graphs or long paths. * **Comparison:** PathNNs represent a significant advancement by providing a path-centric approach to surpass WL limits, offering a distinct mechanism compared to higher-order GNNs which operate on k-tuples.

Another notable approach challenging the WL barrier is the **Random Walk with Unifying Memory (RUM) neural network** (1). *Context:* RUM aims to jointly remedy the limited expressiveness (beyond WL), over-smoothing, and over-squashing problems inherent in convolutional* GNNs. * **Mechanism:** By processing random walk trajectories using RNNs and incorporating "anonymous experiment" features that encode topological environments, RUM captures structural information in a fundamentally different way than message passing. * **Theoretical Evidence:** (1) theoretically demonstrates that RUM is more expressive than the 1-WL test, capable of distinguishing non-isomorphic graphs that 1-WL equivalent GNNs cannot. This is validated empirically on synthetic datasets designed to test expressiveness (e.g., distinguishing cycle sizes). * **Implications:** RUM’s non-convolutional nature and reliance on random walk trajectories provide a novel avenue for achieving higher expressiveness, suggesting that moving beyond the direct message-passing paradigm can unlock greater discriminative power.

The theoretical understanding of GNN expressiveness, largely framed by the WL test, has been instrumental in guiding architectural innovations. It highlights a recurring tension: simpler, more scalable GNNs often sacrifice expressiveness, while more powerful models tend to incur higher computational costs. This tension is further complicated by the need for robust empirical validation. As emphasized by (5), even highly expressive models require rigorous benchmarking with realistic evaluation settings (e.g., hard negative sampling for link prediction) to truly assess their practical utility. The ongoing quest for GNNs that are simultaneously expressive, efficient, and scalable remains a central challenge, with the WL test serving as a constant theoretical compass for progress.

3. Enhancing Expressiveness and Overcoming Early Limitations

The foundational Graph Neural Network (GNN) architectures, as discussed in Section 2, laid the groundwork for learning representations on graph-structured data by extending the principles of deep learning to non-Euclidean domains. Models like Graph Convolutional Networks (GCNs) (Wu et al., 2019) and Graph Attention Networks (GATs) (Veličković et al., 2018) demonstrated remarkable success in tasks such as node classification and link prediction. However, these early models, largely rooted in the message-passing paradigm, quickly encountered inherent limitations that constrained their representational power, robustness, and applicability to the full spectrum of real-world graph complexities. A critical theoretical bottleneck, as established by (Schwartz et al., 2018) and (Wang et al., 2019), was their equivalence to the 1-Weisfeiler-Leman (1-WL) graph isomorphism test, which severely restricted their ability to distinguish between structurally distinct non-isomorphic graphs. This limitation meant that many GNNs could not capture intricate structural patterns, such as specific cycle counts or higher-order motifs, which are crucial for understanding complex systems (Kipf & Wainwright, 2016; Schlichtkrull et al., 2018).

Beyond expressiveness, two significant practical challenges emerged as the field sought to apply GNNs to more complex and larger datasets. Firstly, the phenomenon of **over-smoothing** became a pervasive issue, particularly when attempting to build deeper GNN architectures (Kipf & Wainwright, 2016; Schlichtkrull et al., 2018). As information propagated through multiple layers via repeated neighborhood aggregation, node representations tended to converge, becoming indistinguishable and losing their unique discriminative power. This directly contradicted the success of deep neural networks in other domains and fundamentally limited the effective depth of GNNs. Secondly, the implicit **homophily assumption**, where connected nodes are expected to be similar, proved to be a major impediment for real-world graphs exhibiting **heterophily** (Zhu et al., 2020; Sun et al., 2020). In such graphs, connected nodes often possess dissimilar features or labels (e.g., a student connected to a professor in a social network), and traditional GNNs, designed to smooth features across neighbors, performed poorly.

This section delves into the crucial advancements that have pushed GNNs beyond these

initial limitations, marking a significant intellectual evolution in the field. It explores the development of sophisticated mechanisms designed to enhance representational power, moving beyond the 1-WL test by incorporating higher-order information or entirely novel non-convolutional paradigms. A substantial focus is placed on the diverse strategies developed to tackle the over-smoothing problem, ranging from architectural modifications to more fundamental re-thinking of information dynamics, thereby enabling the construction of deeper, more effective GNNs. Furthermore, the section addresses the critical challenge of adapting GNNs to diverse graph structures, particularly those exhibiting heterophily, moving beyond the traditional homophily assumption to handle the inherent complexities of real-world networks. This progression reflects a field grappling with the fundamental trade-offs between expressiveness, efficiency, and generalizability, continuously seeking to build models that are not only powerful but also robust and adaptable across a myriad of applications, from recommender systems (4) to computer vision (7). The innovations discussed here represent a collective effort to unlock the full potential of GNNs, transforming them from promising initial models into versatile and robust tools capable of tackling the intricate challenges posed by complex graph data.

Higher-Order and Advanced Message Passing Mechanisms

The inherent limitations of early Graph Neural Networks (GNNs) in capturing complex structural information, primarily due to their equivalence to the 1-Weisfeiler-Leman (1-WL) graph isomorphism test, became a significant bottleneck for their representational power (122; 65). As discussed in Section 2, models like GCNs (?) and GATs (?) struggle to distinguish between certain non-isomorphic graphs, such as regular graphs with the same degree sequence but different cycle structures (60). This fundamental constraint spurred extensive research into designing higher-order and advanced message-passing mechanisms that could surpass the 1-WL barrier, thereby enhancing GNNs’ expressiveness and their ability to learn richer, more discriminative graph representations.

One prominent family of approaches to enhance expressiveness involves directly mimicking **higher-order Weisfeiler-Leman (k-WL) tests**. * **Context:** The motivation

here is to overcome the 1-WL limitation by considering not just individual nodes and their immediate neighbors, but larger structural units like k-tuples of nodes. * **Mechanism:** These "k-GNNs" or "k-WL GNNs" operate on sets of k-nodes, where messages are passed between these k-tuples. For example, a 2-WL GNN would consider pairs of nodes and their relationships, allowing it to distinguish graphs that 1-WL cannot, such as non-isomorphic regular graphs (65). The aggregation function is extended to combine information from neighboring k-tuples. * **Core Innovation:** The direct increase in the theoretical discriminative power by operating on more complex structural primitives (k-tuples) rather than just single nodes. * **Theoretical Limitations:** While theoretically more expressive, achieving k-WL power comes at a steep computational cost. The number of k-tuples grows polynomially with the number of nodes (N^k), making these models prohibitively expensive for large graphs even for small k . This scalability issue is a significant practical barrier, limiting their widespread adoption. * **Practical Limitations:** The increased complexity also makes these models harder to implement and train, requiring specialized data structures and aggregation schemes. * **Comparison:** These methods directly address the expressiveness problem by elevating the "resolution" of graph comparison, but they introduce a severe trade-off with computational efficiency. This exemplifies a recurring tension in GNN research: the pursuit of higher expressiveness often comes at the expense of scalability.

A distinct and increasingly influential approach to enhancing expressiveness focuses on incorporating **path information**. Paths inherently encode richer, non-local structural relationships that go beyond immediate neighborhood aggregates. * **Context:** Building on the understanding that local message passing is insufficient for complex structural patterns, researchers sought to leverage longer-range dependencies. * **Method: Path Neural Networks (PathNNs)**, introduced by (6), exemplify this direction. PathNNs update node representations by aggregating information from various paths emanating from each node. * **Mechanism:** For each path length, a recurrent layer (e.g., RNN) is used to encode paths into vectors. The crucial innovation lies in operating on "annotated sets of paths" ($\tilde{S}P$, $S\tilde{P}+$, $\tilde{A}P$). Nodes within these paths are recursively annotated with

hashes of their respective annotated path sets of shorter lengths. This creates a richer, hierarchical structural encoding that captures context beyond simple feature aggregation. For instance, the All Simple Paths (\tilde{AP}) variant aggregates all simple paths up to a fixed length K . * **Evidence:** (6) formally demonstrated that **AP-Trees** are strictly more powerful than **WL-Trees**. More importantly, PathNN variants operating on annotated sets of paths are strictly more powerful than the 1-WL algorithm. The most expressive variant, \tilde{AP} , was empirically shown to distinguish graphs that are even 3-WL indistinguishable, a significant leap in discriminative power. This was validated on synthetic datasets designed to test expressive power and on real-world graph classification and regression tasks, where PathNNs achieved high performance. * **Theoretical Limitations:** While powerful, finding all simple paths in a graph is NP-hard. PathNNs address this by fixing a maximum path length K , which is a practical constraint. The theoretical proofs for expressiveness rely on the recursive annotation scheme, which is computationally intensive for long paths or dense graphs. * **Practical Limitations:** The scalability of the \tilde{AP} variant can be challenging, as the number of simple paths can grow exponentially with length K and graph density. This limits the practical maximum path length that can be effectively used. * **Comparison:** PathNNs offer a compelling, path-centric alternative to k-WL GNNs. While both aim for higher expressiveness, PathNNs achieve this by explicitly modeling sequential structural information, providing a different lens than k-tuples. This approach highlights a methodological trend towards incorporating more explicit structural knowledge into GNN architectures, moving beyond implicit aggregation.

A more radical departure from the traditional message-passing paradigm is the development of **non-convolutional GNNs**. These models fundamentally rethink how information is propagated and aggregated, aiming to overcome not only expressiveness limitations but also other pathologies like over-smoothing and over-squashing. ***Context:** The **Random Walk with Unifying Memory (RUM) neural network** (1) directly confronts the fundamental technical problems of convolution-based* GNNs: limited expressiveness (beyond 1-WL), over-smoothing, and over-squashing (difficulty learning long-range dependencies due to information bottleneck). RUM argues that convolution-based*

GNNs inherently suffer from these issues, necessitating a completely different approach.

Mechanism: *RUM is an entirely convolution-free architecture. Instead of aggregating messages from direct neighbors, it stochastically samples finite-length random walks for each node. An RNN (specifically a GRU) then processes the sequence of node embeddings along these walks. This RNN merges "semantic representations" (node features) with "anonymous experiment" features, which describe the topological environment** of a walk by recording the first unique occurrence of a node. Node representations are then formed by averaging the unifying memories of walks terminating at that node, and graph representations are summed from these node representations. * **Core Innovation:** The "anonymous experiment" is a novel method for encoding topological context, and the use of RNNs to process random walk trajectories provides a fundamentally different and non-local information propagation mechanism. The architecture is designed to jointly remedy the three pathologies. * **Theoretical Contributions:** (1) theoretically demonstrates that RUM is more expressive than the 1-WL test, capable of distinguishing non-isomorphic graphs that 1-WL equivalent GNNs cannot (e.g., cycle sizes, radius). This is a significant theoretical claim, validated empirically on synthetic datasets. Furthermore, RUM is theoretically and experimentally shown to attenuate over-smoothing (its expected Dirichlet energy does not diminish even with long walks) and alleviate over-squashing (slower decay in inter-node Jacobian). * **Practical Advantages:** RUM is shown to be faster and more scalable than existing convolutional and many walk-based GNNs, with a runtime complexity of $O(|V|lkD)$ that is agnostic to the number of edges $|E|$. It is naturally compatible with mini-batching. * **Theoretical Limitations:** The theoretical proofs for expressiveness and over-smoothing alleviation rely on assumptions such as universal and injective functions for the internal mappings (ϕ_x, ϕ_u, f) and connected, unweighted, undirected graphs. While alleviating over-squashing, RUM does not fully solve the information bottleneck with exponentially growing reception fields but rather improves the gradient flow. * **Comparison:** RUM represents a significant paradigm shift, offering a compelling alternative to convolution-based GNNs by jointly addressing expressiveness, over-smoothing, and over-squashing through a non-convolutional, random-walk-based ap-

proach. This contrasts with methods that incrementally improve message passing or rely on higher-order WL tests. It highlights a convergent research direction towards models that can capture global structural information more effectively and efficiently.

In summary, the quest for higher expressiveness in GNNs has led to diverse and innovative solutions. From the direct, but computationally intensive, approach of k-WL GNNs to the path-centric and recursively annotated PathNNs, and finally to the entirely non-convolutional RUM, the field has moved beyond the strict confines of 1-WL equivalence. These advancements underscore a critical tension: how to achieve high discriminative power without incurring prohibitive computational costs. The shift towards incorporating more explicit structural information (paths, random walks) and rethinking the fundamental propagation mechanisms represents a mature understanding of the limitations of early, simplistic message-passing models. This continuous evolution is crucial for GNNs to tackle the increasingly complex and diverse graph learning tasks encountered in real-world applications (4; 7).

Tackling Over-smoothing and Enabling Deep GNNs

The promise of deep learning lies in its ability to learn hierarchical representations through multiple layers of non-linear transformations. However, for Graph Neural Networks (GNNs), this promise has been significantly hampered by the pervasive problem of **over-smoothing** (115; 81). As discussed in Section 2, over-smoothing occurs when, after several layers of message passing, the representations of all nodes in a connected component converge to a single, indistinguishable vector. This loss of discriminative power fundamentally limits the effective depth of GNNs, contrasting sharply with the success of very deep architectures in domains like computer vision (e.g., ResNets with hundreds of layers). Overcoming over-smoothing is paramount for building truly deep and powerful GNNs capable of capturing long-range dependencies and complex hierarchical features.

Several strategies have emerged to tackle over-smoothing, broadly falling into categories of architectural modifications, regularization techniques, and more fundamental re-thinking of information dynamics.

1. Architectural Modifications and Skip Connections: * **Context:** Inspired by the success of residual connections in deep Convolutional Neural Networks (CNNs) (?), early attempts to deepen GNNs focused on similar architectural enhancements. The core idea is to allow information from initial layers to bypass intermediate layers, preventing complete feature homogenization. * **Mechanism:** * **Residual Connections:** Adding the input of a layer to its output, often after a transformation, helps propagate initial features and stabilize gradients. DeepGCNs (?) and GCNII (?) are prime examples. GCNII, for instance, uses an "initial residual" connection that adds a weighted sum of the initial node features to the output of each layer, alongside a "identity mapping" that adds the previous layer's output. This allows the model to retain information from the input layer, mitigating the rapid convergence of features. * **Skip Connections:** More generally, these allow features from earlier layers to be directly fed into later layers or the final readout layer. This helps preserve local information that might otherwise be smoothed out. * **Normalization Layers:** Techniques like Layer Normalization or Batch Normalization are also employed to stabilize activations and prevent features from collapsing. * **Core Innovation:** Adapting established deep learning principles to GNNs to facilitate deeper architectures. * **Evidence:** These methods have enabled the training of significantly deeper GNNs. For example, (52) demonstrated the ability to train GNNs with up to 1000 layers using residual connections, achieving competitive performance. Similarly, (18) proposed decoupling the depth and scope of GNNs, allowing for deeper models without necessarily increasing the receptive field size, which can also contribute to over-smoothing. ***Theoretical Limitations:** While these techniques alleviate over-smoothing, they often do not fundamentally prevent* the feature convergence problem. The underlying message-passing mechanism still tends to smooth features, and skip connections merely provide a bypass, not a cure. The Dirichlet energy, a measure of feature smoothness, still tends to decrease, albeit at a slower rate (117). * **Practical Limitations:** Even with skip connections, very deep GNNs can still be challenging to train due to vanishing/exploding gradients and increased computational cost.

2. Regularization and Graph Rewiring: * **Context:** Over-smoothing is a con-

sequence of excessive information propagation and aggregation. Regularization aims to disrupt this process or modify the graph structure to prevent rapid feature mixing. *

Mechanism: *

- * **DropEdge:** (?) proposed randomly dropping edges during training. This acts as a stochastic regularization technique, preventing any single path from dominating information flow and forcing the model to rely on diverse paths. It effectively creates a slightly different graph at each training step, reducing the over-smoothing effect.
- * **Graph Rewiring:** This involves dynamically or statically modifying the graph structure. For instance, GPR-GNN (114) uses personalized PageRank to adaptively combine features from different hops, which can be seen as a form of implicit rewiring. Other methods dynamically add or remove edges based on learned node similarities or other criteria (97).
- * **Core Innovation:** Introducing stochasticity or modifying graph topology to control information flow and prevent feature homogenization.
- * **Evidence:** DropEdge has been shown to improve performance and enable deeper GNNs by mitigating over-smoothing. Graph rewiring techniques can also improve robustness and performance on specific tasks.
- * **Theoretical Limitations:** DropEdge can reduce the overall information flow, potentially hindering the learning of long-range dependencies if too many edges are dropped. Graph rewiring can be computationally expensive and may alter the original graph semantics, requiring careful justification.
- * **Practical Limitations:** The optimal dropout rate for DropEdge can be dataset-dependent. Dynamic rewiring methods can add significant overhead to training.

3. Decoupled Propagation and Transformation: *

- * **Context:** Traditional GNN layers combine feature transformation and propagation. Decoupling these steps can offer more control over the smoothing process.
- * **Mechanism:** Approaches like APPNP (114) and SGC (?) separate the non-linear feature transformation from the linear propagation step. Initial node features are transformed once, and then these transformed features are propagated linearly across the graph for multiple steps. GCNII (?) also leverages this idea by propagating initial features through a linear combination of previous layer’s features and initial features.
- * **Core Innovation:** Preventing repeated non-linear transformations from rapidly mixing features across layers, allowing for deeper propagation

without immediate over-smoothing. This also allows for decoupling the "depth" (number of propagation steps) from the "scope" (number of non-linear transformations) (18). *

Evidence: These models demonstrate improved performance and deeper architectures compared to standard GCNs, particularly on tasks benefiting from extensive propagation.

* **Theoretical Limitations:** The reliance on linear propagation after initial transformation might limit the model's capacity to learn complex, non-linear interactions across multiple hops.

* **Practical Limitations:** The optimal number of propagation steps needs to be tuned, and too many steps can still lead to over-smoothing.

4. Fundamental Re-thinking of Information Dynamics: Fractional Calculus and Non-Convolutional GNNs: *

Context: The limitations of existing methods suggest that over-smoothing might be an inherent property of the Markovian, integer-order dynamics assumed by most GNNs. A more fundamental change to the information propagation mechanism is needed.

* **Method Family A: Fractional-Order Graph Neural Networks (e.g., FROND (3)):**

* **Mechanism:** The **FRactional-Order graph Neural Dynamical network (FROND)** framework (3) generalizes continuous GNNs by replacing the integer-order differential operator (which models instantaneous, local changes) with the Caputo fractional derivative.

This allows FROND to inherently integrate the entire historical trajectory of node features into their update process, enabling the capture of non-local and memory-dependent dynamics.

For the fractional linear diffusion model (F-GRAND-l), this can be interpreted as a non-Markovian random walk where the walker's complete path history influences future steps.

* **Core Innovation:** Introducing non-local fractional derivatives to model memory-dependent dynamics in continuous GNNs. This fundamentally alters the dynamics of information flow.

Theoretical Contributions: (3) analytically established that the non-Markovian random walk in FROND leads to a slow algebraic* rate of convergence to stationarity, which inherently mitigates over-smoothing, unlike the exponentially swift convergence in Markovian integer-order models.

This provides a strong theoretical basis for its effectiveness.

* **Evidence:** Fractional adaptations of various continuous GNNs (e.g., F-GRAND, F-GRAND++) consistently demonstrated improved performance compared to their integer-order counterparts

across diverse datasets. * **Theoretical Limitations:** The framework relies on numerical FDE solvers, which may introduce computational considerations. The primary focus is on $\beta \in (0, 1]$ for initial conditions, though broader definitions are mentioned. ***Comparison:*** *FROND offers a principled, theoretically grounded solution to over-smoothing by altering the dynamics** of information flow, rather than just architectural tweaks or regularization. It represents a significant advancement by generalizing existing continuous GNNs and opening new research directions for modeling complex, memory-dependent feature updates.

* **Method Family B: Non-Convolutional GNNs (e.g., RUM (1)):** ***Context:*** *Over-smoothing is identified as a core pathology of convolution-based** GNNs. * **Mechanism:** As detailed in the previous subsection, the **Random Walk with Unifying Memory (RUM) neural network (1)** is an entirely convolution-free architecture that processes random walk trajectories using RNNs. This mechanism fundamentally avoids the local, iterative averaging that leads to over-smoothing in traditional GNNs. * **Core Innovation:** Jointly remedies over-smoothing, limited expressiveness, and over-squashing through a novel non-convolutional paradigm. * **Theoretical Contributions:** RUM is theoretically and empirically shown to attenuate over-smoothing, with its expected Dirichlet energy not diminishing even with long walks, unlike convolutional GNNs. * **Comparison:** RUM represents a radical departure that addresses over-smoothing as a systemic issue of convolution, offering a comprehensive solution alongside enhanced expressiveness. It contrasts with FROND by achieving non-Markovian dynamics through random walks and RNNs, rather than fractional derivatives in continuous models.

In conclusion, tackling over-smoothing is a multi-faceted challenge that has driven significant innovation in GNN architecture and theory. While architectural modifications and regularization techniques provide practical improvements, the field is increasingly exploring more fundamental solutions, such as leveraging fractional calculus for memory-dependent dynamics (3) or entirely abandoning convolution-based message passing in favor of random walk-based approaches (1). This evolution signifies a shift from incremental fixes to a deeper understanding of the underlying mechanisms of information propagation,

paving the way for truly deep and robust GNNs capable of capturing complex, multi-scale graph patterns. The ongoing challenge remains to develop models that are not only deep and expressive but also computationally efficient and scalable for real-world applications (4).

Adapting to Diverse Graph Structures and Heterophily

A critical assumption implicitly embedded in many early Graph Neural Network (GNN) architectures, particularly those based on simple neighborhood aggregation (e.g., GCNs (16)), is **homophily**. This principle posits that connected nodes in a graph tend to share similar features or labels (17, 72). While this assumption holds true for many real-world networks, such as citation networks where linked papers often belong to the same research area, it severely limits the performance of these GNNs on **heterophilous graphs**. In heterophilous networks, connected nodes are frequently dissimilar, or even antagonistic, in their attributes or labels (e.g., protein-protein interaction networks, financial transaction graphs, or social networks connecting individuals with diverse interests) (128; 217). The smoothing effect inherent in standard message passing, which averages or sums neighbor features, becomes detrimental in such scenarios, as it blends dissimilar information, leading to degraded node representations. Adapting GNNs to effectively handle diverse graph structures, especially those exhibiting heterophily, has become a crucial research direction.

1. Explicit Heterophily-Aware Aggregation Mechanisms: * **Context:** The direct smoothing of dissimilar features in heterophilous graphs leads to poor performance. The motivation is to design aggregation functions that can intelligently differentiate between homophilous and heterophilous connections. * **Mechanism:** This family of methods modifies the message-passing scheme to explicitly account for node dissimilarity. * **Separate Aggregation:** Some models propose separate aggregation mechanisms for different types of neighbors. For instance, H2GCN (128) aggregates information from both 1-hop neighbors (which might be heterophilous) and higher-order neighbors (which are more likely to be homophilous due to structural properties), then combines these distinct

representations. * **Adaptive Weighting:** Other approaches introduce learnable weights or attention mechanisms that can assign lower weights to dissimilar neighbors or even negative weights to "repulsive" connections. WRGAT (113) and FAGCN (?) are examples where attention mechanisms are designed to be sensitive to feature differences, allowing the model to adaptively weigh messages. * **High-Pass Filters:** Some methods incorporate high-pass filters alongside traditional low-pass filters to capture feature differences, rather than just similarities (71). * **Core Innovation:** Directly countering the detrimental smoothing effect by making aggregation mechanisms sensitive to node similarity/dissimilarity. * **Evidence:** These models generally demonstrate improved performance on benchmark heterophilous datasets (e.g., Chameleon, Squirrel) compared to traditional GNNs. * **Theoretical Limitations:** Many of these methods still rely on local neighborhood information. Determining the optimal "neighborhood size" or the best way to combine information from different hops can be challenging and dataset-dependent. The explicit estimation of homophily/heterophily can be noisy or difficult. * **Practical Limitations:** These models often require careful tuning of hyperparameters related to aggregation strategies. The computational overhead can also increase due to more complex aggregation functions or multiple aggregation paths.

2. Global Homophily and Adaptive Coefficient Learning (e.g., GloGNN/GloGNN++ (2)): *Context:* While local heterophily-aware aggregations are useful, they may still struggle with adaptively setting personalized optimal neighborhood sizes or leveraging global* homophily where similar nodes might be topologically distant. Existing methods often face prohibitive computational costs for truly global aggregation. *Method:* **GloGNN and GloGNN++ (2)** propose a novel framework that generates node embeddings by aggregating information from all global nodes* in the graph, not just local neighbors. * **Mechanism:** In each layer, a **signed coefficient matrix** $\mathbf{Z}^{(l)}$ is learned, where $\mathbf{Z}^{(l)}_{ij}$ describes the importance of node j to node i . This matrix is derived from an optimization problem inspired by the linear subspace model and is regularized by nodes' multi-hop reachabilities to incorporate both feature and topology similarity. Crucially, $\mathbf{Z}^{(l)}$ allows signed values, enabling it to assign large positive coefficients to homophilous nodes and

small positive or even negative coefficients to heterophilous ones, implicitly combining low-pass and high-pass filtering. **Core Innovation:** *The primary innovation is the concept of **global aggregation with learned, signed coefficients** that adaptively capture correlations between all* nodes, effectively finding global homophily.* A significant technical contribution is the **linear time complexity acceleration** for this global aggregation. By transforming the aggregation equation and reordering matrix multiplications, the time complexity is reduced from $O(N^3)$ or $O(N^2c)$ to $O(k^2N)$ (where k is the number of labels, typically small), making global aggregation computationally feasible for large graphs. *

Theoretical Contributions: (2) provides a theoretical proof that both the learned coefficient matrix $\mathbf{Z}^{(l)}$ and the generated node embedding matrix $\mathbf{H}^{(l+1)}$ exhibit a **grouping effect**. This means nodes with similar features and local structures (even if topologically distant) will have similar coefficient vectors and representation vectors, explaining the model’s effectiveness in leveraging global homophily. *

Evidence: Extensive experiments on 15 benchmark datasets (covering diverse domains, scales, and heterophilies) demonstrated that GloGNN and GloGNN++ achieve superior performance and are highly efficient compared to 11 state-of-the-art GNN models. *

Limitations: The paper primarily focuses on node classification tasks. While efficient, the linear time complexity still depends on the number of labels k , which might be large in some multi-class settings. *

Comparison: GloGNN represents a significant advancement by moving beyond local or multi-hop aggregation to a principled and efficient global approach. It addresses the limitations of previous heterophily methods by adaptively learning node correlations across the entire graph, rather than relying on fixed neighborhoods or explicit homophily estimation. This highlights a methodological trend towards more adaptive and globally informed aggregation strategies.

3. Graph Transformation and Rewiring: *

Context: Instead of modifying the GNN architecture, some approaches modify the graph structure itself to make it more amenable to existing GNNs. *

Mechanism: This involves pre-processing the graph to either enhance homophily (e.g., adding edges between similar but disconnected nodes) or create auxiliary structures that capture heterophilous relationships (e.g., meta-paths

in heterogeneous information networks (140; 143)). For instance, virtual nodes can be added to connect distant but related nodes, or edges can be rewired based on feature similarity. * **Core Innovation:** Shifting the burden of handling heterophily from the GNN model to the graph structure itself. * **Evidence:** These methods can improve performance on heterophilous graphs by creating a more "homophilous-friendly" topology. * **Theoretical Limitations:** Graph transformation can be computationally intensive, especially for large graphs. It may also inadvertently alter the original graph semantics or introduce spurious connections, requiring careful validation. * **Practical Limitations:** The design of effective graph transformation rules often requires domain expertise and can be specific to certain graph types.

4. Decoupling Feature Transformation and Propagation: * **Context:** Similar to its role in mitigating over-smoothing, decoupling feature transformation from propagation can also help with heterophily. * **Mechanism:** *By performing non-linear feature transformations locally at each node* before* propagation, and then propagating these transformed features, the model can prevent the immediate mixing of raw, dissimilar features. This allows nodes to learn richer individual representations before interacting with potentially heterophilous neighbors. GCNII (?), with its initial residual connections, implicitly benefits from this by preserving original feature information. * **Core Innovation:** Controlling when and how features are mixed, allowing for initial feature refinement before aggregation. * **Evidence:** Models employing this strategy can show improved robustness on heterophilous graphs. * **Theoretical Limitations:** The extent to which this fully addresses heterophily depends on the power of the initial feature transformation and the subsequent propagation mechanism. * **Practical Limitations:** Still subject to the limitations of the underlying propagation mechanism.

The challenge of adapting GNNs to diverse graph structures, particularly those exhibiting heterophily, has driven the field to move beyond the simplistic homophily assumption. The evolution from local, ad-hoc modifications to more principled, global, and efficient solutions like GloGNN (2) signifies a mature understanding of this problem. This shift is crucial for expanding the applicability of GNNs to a broader range of real-world

scenarios, where heterophily is often the norm rather than the exception. The ongoing debate on whether homophily is a "necessity" or a "curse" for GNNs (72; 217) continues to fuel research into more robust and adaptive graph learning paradigms. Furthermore, the rigorous evaluation of these advanced models, as highlighted by (5) in the context of link prediction, is essential to ensure that innovations truly translate into practical gains across diverse and challenging graph datasets.

4. Advanced Methodologies: Robustness, Generalization, and Efficiency

The evolution of Graph Neural Networks (GNNs) has progressed significantly beyond their foundational message-passing paradigms (as discussed in Section 2) and the initial efforts to enhance expressiveness and depth (Section 3). While these earlier advancements addressed core architectural limitations, the deployment of GNNs in real-world, high-stakes applications necessitates a new generation of sophisticated methodologies. This section delves into these advanced techniques, focusing on three critical dimensions: **robustness**, **generalization**, and **efficiency**. These aspects are paramount for ensuring that GNNs can learn effectively from vast, often unlabeled, and imperfect data, adapt swiftly to novel tasks, withstand malicious attacks, and respect the underlying physical symmetries inherent in many scientific domains.

The journey towards robust, generalizable, and efficient GNNs is characterized by several key intellectual shifts. Firstly, the realization that abundant unlabeled graph data represents an untapped resource has propelled the development of **self-supervised learning (SSL) and pre-training strategies**. These methods enable GNNs to learn powerful, transferable representations without explicit human annotation, addressing the data scarcity problem that often plagues specialized graph tasks. This paradigm shift mirrors the success of pre-trained models in natural language processing and computer vision, aiming to create "foundation models" for graphs. Secondly, as GNNs become more versatile, the need for efficient transfer learning across diverse downstream tasks

has led to the emergence of **prompt-based adaptation**. This approach, often inspired by large language models (LLMs), allows for rapid fine-tuning with minimal task-specific data, and increasingly, facilitates multi-modal integration, bridging the gap between graph structures and rich semantic information from text.

Concurrently, the increasing deployment of GNNs in sensitive areas has brought critical concerns about their **robustness against adversarial attacks and resilience to inherent data imperfections**. Unlike traditional machine learning models, GNNs are uniquely vulnerable to perturbations in both node features and graph topology, demanding specialized defense mechanisms. Real-world graphs are also inherently noisy, incomplete, or biased, requiring methods that can learn reliably despite these imperfections. Finally, for scientific domains such as molecular modeling, materials science, and computational physics, the geometric and physical symmetries of the underlying data are not merely features but fundamental constraints. This has driven the development of **geometric and equivariant GNNs**, which are explicitly designed to respect these symmetries, leading to models that are not only more accurate but also more physically meaningful and generalizable to novel configurations.

Collectively, these advanced methodologies represent a concerted effort to mature GNN technology, moving it from a powerful research tool to a reliable and adaptable solution for complex real-world challenges. They reflect a field grappling with the trade-offs between model complexity, data efficiency, security, and scientific fidelity, continuously pushing the boundaries of what GNNs can achieve.

Self-Supervised Learning and Pre-training Strategies

The success of deep learning in domains like natural language processing (NLP) and computer vision (CV) has been largely attributed to the availability of vast amounts of labeled data and the subsequent development of powerful pre-training paradigms (e.g., BERT, GPT, ImageNet pre-training). In contrast, many real-world graph datasets suffer from limited labeled data, making supervised training of complex GNNs challenging. This scarcity has motivated a significant intellectual shift towards **self-supervised learning (SSL) and pre-training strategies** for GNNs, aiming to learn generalizable and robust

representations from abundant unlabeled graph data.

Context and Motivation: The core motivation for SSL in GNNs is to overcome the bottleneck of labeled data, which is often expensive and time-consuming to acquire for graph-structured information. Early GNNs, as discussed in Section 2, typically relied on supervised training for specific tasks. However, the inherent structural information within graphs, even without explicit labels, provides a rich signal for learning. Pre-training allows GNNs to capture fundamental graph properties, such as node proximity, structural roles, and community structures, which can then be efficiently transferred to various downstream tasks with minimal fine-tuning. This approach aims to enhance generalization capabilities, particularly in low-resource settings, and improve the efficiency of model development (20).

Method Family A: Contrastive Learning on Graphs: * **Problem Solved:** Learning robust node and graph representations by maximizing agreement between different views of the same graph or subgraph, without explicit labels. This addresses the challenge of distinguishing meaningful structural patterns from noise. * **Core Innovation & Mechanism:** Graph contrastive learning (GCL) typically involves creating multiple augmented views of a graph (or its nodes/subgraphs) and training an encoder to maximize the similarity of representations from positive pairs (different views of the same entity) while minimizing similarity for negative pairs (views of different entities). * **Augmentation Strategies:** Common augmentations include edge perturbation (adding/removing edges), node feature masking/dropping, subgraph sampling, and diffusion-based transformations (13). For instance, GRACE (?) generates two correlated views of a graph by randomly corrupting its structure and node attributes, then maximizes agreement between the encodings of common nodes across these views. * **Contrastive Objectives:** InfoNCE loss is widely used, pushing positive pairs closer and negative pairs further apart in the embedding space. * **Evidence:** GCL methods have shown significant improvements in node classification, graph classification, and link prediction tasks, often outperforming purely supervised baselines, especially in semi-supervised or few-shot settings (20; 56). For example, SLAPS (56) improves structure learning for GNNs by

using self-supervision, demonstrating enhanced performance on node classification tasks.

* **Theoretical Limitations:** The effectiveness of GCL heavily depends on the quality of augmentation strategies and the selection of negative samples. Poor augmentations might not capture sufficient invariance, while hard negative sampling can be computationally expensive and challenging. The choice of pretext task (e.g., which augmentations to use) is often heuristic and lacks strong theoretical guarantees for optimal representation learning across all graph types. * **Practical Limitations:** GCL can be computationally intensive, especially for large graphs, due to the need for multiple forward passes for augmented views and large batches for effective negative sampling. The hyperparameter tuning for augmentation strength and temperature parameters can be complex. * **Comparison:** GCL methods, such as GRACE and DGI (?), focus on learning discriminative representations by contrasting different graph views. This contrasts with generative SSL methods (discussed next) that aim to reconstruct corrupted inputs. While both aim for robust representations, GCL often excels in tasks requiring fine-grained distinction between entities.

Method Family B: Generative Pre-training on Graphs: * **Problem Solved:** Learning general-purpose graph representations by reconstructing corrupted graph inputs or predicting missing information. This helps GNNs understand the underlying generative process of graph structures and features. * **Core Innovation & Mechanism:** Generative pre-training tasks involve predicting masked node features, reconstructing masked edges, or generating entire subgraphs. * **Masked Attribute/Edge Prediction:** Similar to masked language modeling (e.g., BERT), GPT-GNN (112) proposes generative pre-training of GNNs by masking a portion of node features or edges and training the GNN to reconstruct them. This forces the model to learn context-aware representations. * **Graph Generation:** More advanced generative models aim to learn the distribution of graphs themselves, which can then be used for representation learning. * **Evidence:** Generative pre-training, particularly for tasks like link prediction in biomedical networks, has shown promise in learning meaningful representations (231). Mole-BERT (169) rethinks pre-training GNNs for molecules, demonstrating improved performance in molecu-

lar property prediction by leveraging generative tasks. * **Theoretical Limitations:** The choice of what to mask or generate can significantly influence the learned representations. If the masking strategy is too simple, the model might not learn sufficiently complex patterns. If too aggressive, it might struggle to learn coherent structures. * **Practical Limitations:** Generative models can be complex to design and train, especially for graph generation tasks, which often involve discrete structures. The reconstruction loss might not always align perfectly with downstream task objectives. * **Comparison:** Generative pre-training, exemplified by GPT-GNN (112), focuses on reconstructing missing information, thereby learning a more comprehensive understanding of graph structure and features. This is often complementary to contrastive methods; some frameworks even combine both. The evolution here shows a clear trend towards adapting successful NLP pre-training paradigms to the graph domain.

Method Family C: Pre-training for Specific Downstream Tasks: * **Problem Solved:** Tailoring pre-training to improve performance on specific types of downstream tasks (e.g., node classification, graph classification, link prediction). * **Core Innovation & Mechanism:** Instead of general-purpose pretext tasks, some methods design pre-training objectives that are more aligned with the target downstream task. For example, (49) investigates various strategies for pre-training GNNs, including those focused on node-level or graph-level tasks, and finds that pre-training can significantly boost performance. (24) explores learning to pre-train GNNs, optimizing the pre-training process itself. * **Evidence:** Pre-training strategies have been shown to be particularly effective for improving link prediction in biomedical networks (231), where learning rich node and edge representations is crucial. * **Theoretical Limitations:** Task-specific pre-training might lead to less generalizable representations compared to more universal SSL objectives, potentially limiting transferability to very different tasks. * **Practical Limitations:** Requires some prior knowledge about the nature of the downstream tasks, which might not always be available.

Synthesis and Implications: The field of SSL and pre-training for GNNs is rapidly maturing, moving towards more sophisticated pretext tasks and architectural designs. A

critical tension exists between learning highly generalizable representations that transfer across diverse tasks and designing task-specific pre-training objectives that yield superior performance on a narrow set of tasks. The evolution shows a clear shift from simple unsupervised embedding methods to complex, multi-faceted SSL frameworks. For instance, the transition from early unsupervised methods like DeepWalk (Veličković et al., 2016) to sophisticated contrastive methods like GRACE (Zhu et al., 2020) and generative models like GPT-GNN (Zhu et al., 2021) highlights the increasing complexity and effectiveness of pretext tasks. The ultimate goal is to enable GNNs to learn from vast unlabeled data, much like large language models learn from text corpora, thereby facilitating efficient and robust deployment across a wide array of real-world applications.

Prompt-based Adaptation and Multi-modal Learning

The paradigm of pre-training, as discussed in the previous subsection, has enabled GNNs to learn powerful representations from unlabeled data. However, efficiently adapting these pre-trained models to diverse downstream tasks, especially with limited labeled data, remains a significant challenge. Inspired by the success of prompt engineering and prompt tuning in Large Language Models (LLMs), **prompt-based adaptation** has emerged as a promising strategy for GNNs, offering efficient transfer learning. Furthermore, the increasing complexity of real-world data often necessitates **multi-modal learning**, integrating graph structures with other data modalities, most notably text, often mediated by LLMs.

Context and Motivation: Traditional fine-tuning of pre-trained GNNs involves updating all model parameters, which can be computationally expensive and prone to catastrophic forgetting, especially when adapting to many diverse tasks or with limited data. Prompt-based adaptation aims to address this by freezing most pre-trained parameters and only optimizing a small set of "prompt" parameters or by designing specific input prompts that guide the model towards the desired task. This approach significantly improves efficiency and generalization in low-resource settings. Simultaneously, many real-world entities represented in graphs (e.g., scientific papers, social media posts, molecules with textual descriptions) inherently possess multi-modal information. Integrating this

rich semantic context, particularly from LLMs, can enhance GNNs' understanding and predictive power.

Method Family A: Prompt Tuning for GNNs: * **Problem Solved:** Efficiently adapting pre-trained GNNs to new downstream tasks with minimal labeled data and computational cost, without full fine-tuning. This addresses the challenge of transfer learning and resource efficiency. * **Core Innovation & Mechanism:** Prompt tuning for GNNs involves adding task-specific "prompts" to the input or intermediate layers of a pre-trained GNN. These prompts are typically small, learnable vectors that guide the model's behavior for a specific task. * **Soft Prompts:** Instead of human-designed textual prompts, GNN prompt tuning often uses "soft prompts" – continuous vectors that are learned during adaptation. These can be appended to node features, graph embeddings, or even integrated into the message-passing process. * **GPPT (Graph Pre-training and Prompt Tuning):** (15) introduced GPPT, a framework that leverages pre-trained GNNs and prompt tuning. It learns task-specific prompts that are added to the input node features, allowing the pre-trained GNN to adapt to new tasks by only updating these prompt parameters. * **Universal Prompt Tuning:** (75) proposed a universal prompt tuning framework for GNNs, demonstrating its effectiveness across various graph tasks and datasets. This suggests a generalizable mechanism for efficient adaptation. * **GraphPrompt:** (77) unifies pre-training and downstream tasks for GNNs through a prompting mechanism, showcasing improved performance and efficiency. * **Evidence:** Prompt-based methods like GPPT (15) have shown significant performance gains over traditional fine-tuning in few-shot and zero-shot settings, achieving comparable or superior results with substantially fewer trainable parameters and faster adaptation times. (166) further extends this by proposing multi-task prompting for GNNs, enabling a single model to handle multiple tasks efficiently. ***Theoretical Limitations:** The theoretical understanding of why* prompts work so effectively in GNNs is still evolving. The optimal design of prompts (e.g., where to insert them, their dimensionality) is often heuristic. The expressiveness of prompt-tuned GNNs compared to fully fine-tuned models can be limited for highly complex or novel tasks.* * **Practical Limitations:** Designing

effective prompts can be challenging, requiring careful consideration of the task and the pre-trained model’s capabilities. The performance can be sensitive to prompt initialization and hyperparameter choices. * **Comparison:** Prompt tuning offers a more parameter-efficient alternative to full fine-tuning, especially beneficial for deploying large pre-trained GNNs across many tasks. It represents a shift from modifying the entire model to subtly guiding its behavior, a trend seen across deep learning.

Method Family B: Multi-modal Integration with Large Language Models (LLMs): * **Problem Solved:** Enriching graph representations with external semantic knowledge, particularly from text, and enabling GNNs to understand and process multi-modal inputs. This addresses the limitation of GNNs being solely reliant on graph structure and features. * **Core Innovation & Mechanism:** This involves combining the structural reasoning capabilities of GNNs with the powerful semantic understanding of LLMs. * **Text-Enhanced GNNs:** Initial approaches involved using text embeddings (e.g., from BERT) as node features in GNNs (41). More advanced methods directly integrate LLMs into the GNN architecture or use them for prompt generation. * **Hybrid-LLM-GNN:** (287) proposes Hybrid-LLM-GNN, integrating LLMs and GNNs for enhanced materials property prediction. This framework leverages the LLM’s ability to process textual descriptions of materials and combine them with the GNN’s understanding of molecular graphs. * **Weak Text Supervision:** (73) explores whether GNNs can learn language with extremely weak text supervision, demonstrating the potential for implicit multi-modal learning. * **Prompting LLMs for Graph Tasks:** LLMs can be prompted to generate graph structures, node features, or even provide reasoning for GNN predictions, effectively acting as a knowledge base or reasoning engine for graph tasks. * **Evidence:** Multi-modal GNNs, particularly those integrating LLMs, have shown improved performance on tasks requiring a blend of structural and semantic understanding, such as knowledge graph completion, document classification, and drug discovery (287; 239). For example, combining LLMs with GNNs for materials property prediction can leverage both the chemical structure and textual descriptions of synthesis methods or properties, leading to more accurate predictions. * **Theoretical Limitations:** Aligning

the distinct representation spaces of graphs and text (or other modalities) is a complex challenge. The "black-box" nature of LLMs can make the combined model less interpretable. The causal relationships between textual descriptions and graph properties are often implicit and hard to model explicitly. * **Practical Limitations:** Integrating LLMs with GNNs can be computationally very expensive, requiring significant memory and processing power. The choice of how to fuse information (early, late, or hybrid fusion) is an active research area and can significantly impact performance. Data heterogeneity and potential biases in either modality can propagate and amplify. * **Comparison:** This represents a convergence of two powerful AI paradigms. Unlike traditional GNNs that operate solely on graph data, multi-modal GNNs leverage external knowledge sources, particularly text, to enrich their understanding. This is a significant step towards more holistic and intelligent graph AI.

Synthesis and Implications: Prompt-based adaptation and multi-modal learning are driving GNNs towards greater efficiency, generalizability, and intelligence. The evolution from full fine-tuning to parameter-efficient prompt tuning, and from single-modality graph processing to multi-modal integration with LLMs, signifies a maturation of the field. This trend is crucial for enabling GNNs to operate effectively in complex, data-rich, and resource-constrained real-world environments. The ability to efficiently transfer knowledge and integrate diverse information sources positions GNNs as key components in future intelligent systems, particularly in domains where structured relationships intersect with rich semantic content.

Robustness to Adversarial Attacks and Data Imperfections

As Graph Neural Networks (GNNs) are increasingly deployed in critical applications, ranging from financial fraud detection (246) and cybersecurity (21; 191; 321) to recommender systems (4) and drug discovery (274), their vulnerability to malicious attacks and sensitivity to inherent data imperfections become paramount concerns. Unlike traditional deep learning models, GNNs are susceptible to perturbations in both node features and graph topology, making robustness a multi-faceted challenge. This section explores methodologies developed to enhance GNN resilience against adversarial attacks and their

ability to handle real-world data imperfections.

Context and Motivation: The message-passing mechanism, which aggregates information from neighbors, makes GNNs particularly vulnerable. A small perturbation to a single node’s features or a few edges can propagate throughout the graph, drastically altering node representations and model predictions (87). This vulnerability can be exploited by adversarial attacks, where an attacker intentionally crafts perturbations to degrade GNN performance or induce misclassifications. Furthermore, real-world graphs are rarely perfect; they often contain noise, missing links, spurious connections, or inherent biases (11). Ensuring GNN trustworthiness requires addressing both intentional attacks and unintentional imperfections (61; 66).

Method Family A: Defenses Against Adversarial Attacks: * **Problem Solved:** Protecting GNNs from malicious perturbations to node features or graph structure, ensuring reliable predictions even under attack. This addresses the security and trustworthiness of GNN deployments. * **Core Innovation & Mechanism:** Adversarial attacks on GNNs can be broadly categorized into poisoning attacks (modifying the training graph) and evasion attacks (modifying the test graph). Defenses aim to either make the GNN inherently robust or detect and mitigate attacks. * **Adversarial Training:** Inspired by defenses in image classification, adversarial training for GNNs involves augmenting the training data with adversarially perturbed graphs. (82) explores adversarial training for GNNs, demonstrating its effectiveness in improving robustness against various attack types. This forces the model to learn robust features that are invariant to small perturbations. * **Robust Aggregation Mechanisms:** Some defenses modify the message-passing aggregation to be more resilient. For example, GNNGuard (83) introduces an attention-based mechanism to identify and filter out suspicious messages from adversarial neighbors, effectively preventing the propagation of corrupted information. Other methods use robust aggregation functions (e.g., median instead of mean) or prune unreliable edges. * **Graph Structure Learning (GSL) for Robustness:** Instead of relying on a fixed, potentially compromised graph, GSL methods learn a more robust graph structure alongside node representations. (97) proposes learning graph structure for robust GNNs, where

the model adaptively modifies the adjacency matrix to enhance robustness against adversarial attacks and noise. This can involve adding or removing edges based on feature similarity or other criteria. **Certified Robustness:** *More recently, research has focused on providing* provable guarantees* of robustness. GNNCert (262) offers deterministic certification of GNNs against adversarial perturbations, providing a formal upper bound on the impact of attacks. This is a significant step towards truly trustworthy GNNs. *

Evidence: Adversarial training and robust aggregation methods have shown improved accuracy under attack scenarios, with GNNGuard (83) demonstrating superior performance against various topology attacks. (36) critically evaluates existing GNN defenses, finding that many are not as robust as claimed, highlighting the ongoing challenge. (306) focuses on bounding the expected robustness, providing theoretical insights into defense mechanisms. *

Theoretical Limitations: Adversarial training often provides robustness only against the specific attack types seen during training. Certified robustness methods are typically computationally expensive and may only apply to specific GNN architectures or perturbation types. The trade-off between robustness and utility (clean accuracy) is a recurring tension. *

Practical Limitations: Implementing adversarial training can be complex and computationally intensive. Robust aggregation mechanisms might introduce additional parameters or computational overhead. The effectiveness of defenses can vary significantly depending on the attack model (e.g., poisoning vs. evasion, targeted vs. untargeted) and the graph properties. (250) even explores whether Large Language Models (LLMs) can improve the adversarial robustness of GNNs, suggesting a multi-modal approach to this complex problem. *

Comparison: Defenses range from reactive (adversarial training) to proactive (robust aggregation, GSL) and provable (certified robustness). The evolution shows a shift from heuristic defenses to more principled and theoretically grounded approaches, though scalability remains a challenge.

Method Family B: Resilience to Data Imperfections and Bias Mitigation: *

Problem Solved: Enabling GNNs to learn effectively from noisy, incomplete, or biased real-world graph data, ensuring fair and reliable outcomes. This addresses the practical challenges of data quality and ethical AI. *

Core Innovation & Mechanism: Real-

world graphs often suffer from missing node features, erroneous edges, or inherent biases (e.g., demographic biases in social networks).

- * **Learning with Noisy Graphs and Sparse Labels:** (40) addresses the challenge of robust GNNs for noisy graphs with sparse labels, proposing methods to learn reliable representations despite data imperfections. This often involves imputation techniques, noise-aware training objectives, or adaptive graph filtering.
- * **Debiasing and Fairness:** GNNs can inadvertently learn and amplify biases present in the training data, leading to unfair predictions.
- * **EDITS:** (11) proposes EDITS to model and mitigate data bias for GNNs, focusing on identifying and correcting biased information propagation.
- * **Disentangled Causal Substructure:** (25) debiases GNNs by learning disentangled causal substructures, aiming to separate causal features from confounding biases.
- * **Re-balancing and Sensitive Attribute Leakage:** (12) rethinks fair GNNs from a re-balancing perspective, while (116) focuses on mitigating sensitive attribute leakage to improve fairness. These methods often involve re-weighting samples, adjusting aggregation, or learning fair representations.
- * **Graph Rewiring and Preprocessing:** (257) explores graph rewiring and preprocessing for GNNs based on effective resistance, aiming to create more robust and informative graph structures that are less susceptible to noise and imperfections.
- * **Unnoticeable Backdoor Attacks:** Beyond simple noise, (109) explores unnoticeable backdoor attacks on GNNs, where a trigger can induce a specific misbehavior without being easily detected. This highlights the need for defenses against subtle, targeted data imperfections.
- * **Evidence:** Debiasing methods have shown significant improvements in fairness metrics (e.g., demographic parity, equal opportunity) while maintaining competitive utility. Learning with noisy labels or graphs has demonstrated improved performance compared to naive training on imperfect data. (318) provides a comprehensive benchmark for GNNs under label noise, highlighting the challenges and progress in this area.
- * **Theoretical Limitations:** Defining and measuring fairness in graph contexts is complex and often involves trade-offs between different fairness notions and utility. The underlying causal mechanisms of bias are often hard to model explicitly.
- * **Practical Limitations:** Debiasing techniques can be computationally intensive and may require sensitive attribute information, which might not always

be available or ethically permissible to use. The effectiveness of methods for noisy data depends on the nature and extent of the noise. * **Comparison:** This area shows a clear trend towards building trustworthy GNNs that are not only accurate but also fair and resilient to real-world data challenges. It moves beyond simply improving accuracy to addressing the ethical and practical implications of GNN deployment.

Synthesis and Implications: The pursuit of robustness and resilience is transforming GNN research, moving it towards a more holistic understanding of trustworthiness. The evolution from basic adversarial attacks (104) to sophisticated backdoor attacks (109) and from heuristic defenses to certified robustness (262) reflects the increasing maturity and complexity of the problem. Similarly, the shift from merely handling noisy data to actively mitigating bias and ensuring fairness (12) underscores the field’s growing awareness of societal impact. These efforts are crucial for the widespread adoption of GNNs in sensitive real-world applications, where trust and reliability are paramount.

Geometric and Equivariant Graph Neural Networks

For many scientific and engineering domains, such as molecular biology, materials science, and computational physics, data inherently possesses geometric structure and adheres to specific physical symmetries. Traditional GNNs, while powerful in capturing topological relationships, often struggle to explicitly incorporate or respect these fundamental symmetries (e.g., rotation, translation, reflection invariance or equivariance). This limitation can lead to models that are less accurate, less data-efficient, and less generalizable to novel configurations. The development of **geometric and equivariant GNNs** represents a critical advancement, designing models that intrinsically understand and leverage these symmetries, thereby revolutionizing scientific discovery with GNNs.

Context and Motivation: In molecular modeling, for instance, the energy of a molecule is invariant to its rigid body rotation or translation in space. A GNN predicting molecular properties should ideally exhibit this same invariance. If a GNN is not rotationally invariant, it might predict different properties for the same molecule simply because it is presented in a different orientation, requiring extensive data augmentation or leading to poor generalization. Similarly, in fields like computational mechanics, models should

be equivariant to transformations, meaning that if the input (e.g., a force field) is rotated, the output (e.g., displacement) should also rotate consistently. Early GNNs, primarily focused on graph isomorphism (as discussed in Section 3 regarding WL tests), did not explicitly encode these continuous geometric symmetries, creating a gap for applications where spatial information and physical laws are crucial (226).

Method Family A: Equivariant GNNs for 3D Data: * **Problem Solved:** Designing GNNs that respect the fundamental symmetries of 3D space (e.g., rotation, translation, reflection), leading to models that are inherently invariant or equivariant to these transformations. This addresses the challenge of building physically consistent and data-efficient models for geometric data. * **Core Innovation & Mechanism:** Equivariant GNNs ensure that if the input graph (e.g., a molecule’s atomic coordinates and features) undergoes a geometric transformation, the output representation transforms predictably (equivariance) or remains unchanged (invariance). * **E(n)-Equivariant GNNs:** (102) introduced E(n)-equivariant GNNs, which are equivariant to the Euclidean group E(n) (rotations, translations, reflections). These models achieve equivariance by carefully designing message-passing operations that operate on vectors and tensors in a way that preserves their transformation properties. For example, messages might be constructed using dot products of vectors (which are invariant) or by linearly transforming vectors based on relative positions (which maintain equivariance). * **E(3)-Equivariant GNNs:** Specifically for 3D molecular data, E(3)-equivariant GNNs are crucial. (124) developed E(3)-equivariant GNNs for data-efficient and accurate interatomic potentials, demonstrating that by building in these symmetries, models can learn from significantly less data and achieve higher accuracy in predicting physical properties. GemNet (99) is another example, proposing universal directional GNNs for molecules that explicitly incorporate directional information and symmetries. * **Geometric Message Passing:** These GNNs often incorporate 3D coordinates and distances directly into the message-passing mechanism. Messages are not just scalar features but can be vectors or tensors that transform appropriately under rotations. * **Evidence:** E(3)-equivariant GNNs have achieved state-of-the-art performance in predicting molecular energies, forces, and other properties, often

with significantly less training data than non-equivariant models (124; 99). For example, (124) showed that their $E(3)$ -equivariant GNNs can achieve chemical accuracy with orders of magnitude less data than traditional methods. (51) and (175) provide comprehensive reviews and benchmarks for GNNs in materials science and chemistry, highlighting the impact of these geometric approaches. * **Theoretical Limitations:** Designing truly universal equivariant layers that can handle arbitrary symmetries and tensor representations can be complex. The computational cost of handling higher-order tensors can be substantial, especially for large systems. The expressive power of geometric GNNs is also an active research area (14). * **Practical Limitations:** Implementing equivariant GNNs requires specialized knowledge of group theory and tensor algebra. The choice of which symmetries to enforce and how to represent features (scalars, vectors, tensors) can impact performance and complexity. Scalability to very large molecules or materials systems remains a challenge. (285) questions whether high-degree representations are always necessary in equivariant GNNs, suggesting a potential trade-off. * **Comparison:** Equivariant GNNs represent a paradigm shift from GNNs that are merely invariant to node permutations to those that are also invariant/equivariant to continuous geometric transformations. This contrasts with traditional GNNs that might learn these symmetries implicitly through data augmentation, which is less efficient and less robust.

Method Family B: Positional and Structural Encodings for Geometric Information: * **Problem Solved:** Enhancing standard GNNs with explicit spatial or structural information when full equivariance might be overkill or computationally prohibitive. This addresses the need to incorporate geometric context without a complete architectural overhaul. * **Core Innovation & Mechanism:** These methods add positional or structural encodings to node features or edges, allowing permutation-invariant GNNs to implicitly capture some geometric information. * **Positional Encodings:** (100) explores GNNs with learnable structural and positional representations, adding information about a node’s position within the graph (e.g., shortest path distances, eigenvalues of graph Laplacian) to its features. (23) specifically focuses on equivariant and stable positional encoding for more powerful GNNs, ensuring that these encodings are robust to

small perturbations and respect symmetries. * **Distance-based Features:** For molecular graphs, interatomic distances are often used as edge features or incorporated into attention mechanisms, providing crucial geometric context. * **Geometric Graph Neural Networks:** (254) introduces descriptor-free collective variables from geometric GNNs, showcasing how GNNs can learn complex geometric features without explicit hand-crafted descriptors. * **Evidence:** Incorporating positional encodings or distance-based features has been shown to improve the performance of standard GNNs on tasks requiring geometric awareness, such as molecular property prediction and 3D point cloud processing. * **Theoretical Limitations:** While improving performance, these methods do not provide the same strong theoretical guarantees of equivariance as dedicated equivariant GNNs. They rely on the GNN to implicitly learn how to use these encodings to respect symmetries. * **Practical Limitations:** The choice of positional encoding (e.g., Laplacian eigenvectors, random walk features) can impact performance and scalability. For very large graphs, computing certain positional encodings can be expensive. * **Comparison:** This approach is often seen as a practical compromise between the simplicity of standard GNNs and the theoretical rigor of fully equivariant GNNs. It allows for leveraging existing GNN architectures while injecting crucial geometric information.

Synthesis and Implications: Geometric and equivariant GNNs represent a significant leap forward in applying GNNs to scientific and engineering problems. The evolution from implicitly learning symmetries to explicitly encoding them into the architecture reflects a deeper understanding of the underlying physics and geometry of the data. This has led to models that are not only more accurate and data-efficient but also more interpretable in a physical sense. The ongoing research in this area, including the exploration of symplectic GNNs (327) and the expressive power of geometric GNNs (14), promises to further accelerate discovery in fields like drug design, materials discovery, and computational chemistry, where GNNs are becoming indispensable tools.

5. Trustworthy GNNs: Explainability, Fairness, and Privacy

The widespread adoption of Graph Neural Networks (GNNs) across sensitive domains, from healthcare and finance to social media analysis and cybersecurity, has elevated the concept of "trustworthiness" from a desirable feature to an indispensable requirement. While previous sections have explored the architectural foundations (Section 2), expressive power (Section 3), and robustness against adversarial attacks (Section 4) of GNNs, these advancements alone are insufficient for responsible deployment. Trustworthy GNNs must not only perform accurately and robustly but also be **explainable** (allowing humans to understand their decisions), **fair** (mitigating biases and ensuring equitable outcomes), and **private** (protecting sensitive information). This section delves into the burgeoning research dedicated to these three critical pillars of trustworthy GNNs, recognizing their paramount importance for fostering confidence and ethical use in real-world scenarios (61; 66).

The pursuit of trustworthy GNNs represents a significant intellectual shift in the field, moving beyond purely performance-driven metrics to encompass ethical, societal, and practical considerations. Early GNN research primarily focused on enhancing predictive accuracy and scalability, often treating models as black boxes. However, the realization that GNNs can perpetuate and even amplify biases present in training data, make opaque decisions in critical applications, or inadvertently leak sensitive information, has spurred a concerted effort to imbue them with human-centric qualities. This shift is characterized by several key developments. Firstly, the demand for **Explainable Graph Neural Networks (XGNNs)** has led to diverse methodologies, ranging from post-hoc techniques that pinpoint crucial subgraphs or features for a given prediction to intrinsically interpretable architectures. This addresses the fundamental challenge of understanding *why* a GNN makes a particular decision, a critical step for debugging, auditing, and building user trust.

Secondly, the growing awareness of algorithmic bias has driven extensive research into

fairness and bias mitigation in GNNs. GNNs, by their very nature of propagating information across a graph, can easily spread and reinforce biases embedded in node features or graph topology, leading to discriminatory outcomes for certain demographic groups. Research in this area explores pre-processing techniques to debias input data, in-processing methods that integrate fairness constraints into the training objective, and post-processing adjustments to model outputs. These efforts aim to ensure that GNNs operate equitably, preventing disparate impacts that could arise from training data disparities.

Finally, the inherent interconnectedness of graph data poses unique **privacy challenges**. Sensitive information about individuals (nodes) or their relationships (edges) can be inferred from GNN models or their outputs, even if raw data is not directly exposed. This has necessitated the development of **privacy-preserving GNNs**, employing techniques such as differential privacy, federated learning, and cryptographic methods to protect sensitive information during both training and inference. These advancements are crucial for deploying GNNs in domains like social networks, medical informatics, and financial transactions, where data privacy is a legal and ethical imperative.

Collectively, these three areas are not isolated but often interlinked, presenting complex trade-offs and unresolved tensions. For instance, enhancing privacy might inadvertently reduce model utility or make explanations harder to generate. Similarly, debiasing efforts could impact model accuracy or require access to sensitive attributes, raising privacy concerns. The field is actively grappling with these multifaceted challenges, striving to develop holistic frameworks that can simultaneously optimize for explainability, fairness, and privacy, thereby paving the way for the responsible and ethical deployment of GNN technology in real-world scenarios.

Explainable Graph Neural Networks (XGNNs)

The inherent complexity and black-box nature of deep learning models, including Graph Neural Networks (GNNs), pose significant challenges for their adoption in high-stakes applications where transparency and accountability are paramount. Users and stakeholders

often demand to understand *why* a GNN made a particular prediction, especially when those decisions impact individuals or critical systems. This necessity has given rise to the burgeoning field of Explainable Graph Neural Networks (XGNNs), which aims to shed light on the internal workings and decision-making processes of these models. The development of XGNNs builds upon the foundational GNN architectures discussed in Section 2 and complements the robustness efforts highlighted in Section 4, as understanding failure modes often requires interpretability.

Context and Motivation: Traditional GNNs, with their iterative message-passing mechanisms, aggregate information from complex, non-Euclidean structures, making it difficult to trace the influence of specific input components on the final output. Unlike image or text data, where salient regions or words can be highlighted, graph explanations must consider both node features and the intricate topology. Early attempts at GNNs focused solely on predictive performance, but the demand for trust, debugging capabilities, and compliance with regulations (e.g., GDPR’s “right to explanation”) has spurred a significant research focus on interpretability. The core problem is to provide human-intelligible insights into a GNN’s predictions, which, for graphs, often means identifying crucial subgraphs, nodes, edges, or features (105).

Method Family A: Post-hoc Local Explanations (Subgraph/Feature Importance) * Problem Solved: This family of methods aims to explain individual GNN predictions by identifying the most influential input components (e.g., a critical subgraph, key nodes, or salient features) that led to a specific output. This addresses the need for instance-level transparency, allowing users to understand a single decision. ***Core Innovation & Mechanism:** These approaches typically operate after** a GNN has been trained (post-hoc) and focus on local explanations for a specific input graph. **** GNNExplainer (31):*** One of the pioneering works, GNNExplainer, formulates explanation as an optimization problem. For a given GNN prediction on an input graph, it learns a compact subgraph and a subset of node features that are most influential for that prediction. It achieves this by maximizing the mutual information between the masked subgraph (or features) and the GNN’s prediction, effectively identifying a minimal, yet sufficient,

explanatory subgraph. **SubgraphX (8)**: Building on the intuition that humans often understand graph decisions through coherent substructures, SubgraphX directly identifies important connected* subgraphs. Its core innovation lies in employing Monte Carlo Tree Search (MCTS) to efficiently explore the vast space of possible subgraphs and leveraging Shapley values from cooperative game theory to quantify the importance of each subgraph. Shapley values inherently capture interactions among different graph components, providing a fair and theoretically grounded measure. To address the computational intractability of exact Shapley values, (8) proposes graph-specific approximation schemes, leveraging the L-hop receptive field of GNNs. **CF-GNNExplainer (107)**: This method provides counterfactual explanations by identifying minimal changes to the input graph (e.g., adding/removing edges or changing node features) that would alter the GNN’s prediction to a desired outcome. This offers insights into what would have had to be different* for a different prediction, which can be very informative for understanding decision boundaries. **PGExplainer (38)**: This approach focuses on generating inductive and efficient explanations. Unlike GNNExplainer which trains a separate explainer for each GNN, PGExplainer learns a policy network that can generate explanations for any* GNN, making it more generalizable and efficient for multiple models or unseen graphs. * **GOAt (264)**: Graph Output Attribution (GOAt) explains GNNs via structure-aware interaction index, focusing on how different parts of the graph contribute to the final output, particularly for graph-level tasks. * **Evidence**: GNNExplainer (31) demonstrated the ability to identify human-interpretable subgraphs. SubgraphX (8) empirically showed "significantly improved explanations" and "better explanations for a variety of GNN models" compared to existing methods, providing more coherent and relevant subgraphs. CF-GNNExplainer (107) offers actionable insights by highlighting critical structural dependencies. * **Theoretical Limitations**: The "fidelity" of post-hoc explanations (how accurately they reflect the true model reasoning) is often not guaranteed. Shapley values, while theoretically sound, are approximations in practice for graphs. The definition of a "good" explanation remains subjective and domain-dependent, leading to challenges in rigorous evaluation (43; 76). * **Practical Limitations**: These methods can be computationally

expensive, especially for large graphs or when exact Shapley values are desired. The interpretability of the identified subgraphs can vary, and human users might still struggle with complex patterns. The process often requires tuning specific parameters for the explainer itself. **Comparison:** *SubgraphX (8) distinguishes itself from GNNExplainer (31) by explicitly searching for connected subgraphs and using a game-theoretic approach (Shapley values) for importance, addressing the limitation that GNNExplainer’s identified important nodes/edges might not form a coherent subgraph. PGExplainer (38) further improves efficiency and generalizability by learning an inductive* explainer, contrasting with instance-specific explainers.* The evolution shows a clear trend from simple feature importance to more complex structural (subgraph) and counterfactual explanations, aiming for deeper insights into GNN reasoning.

Method Family B: Intrinsically Interpretable GNNs and Global Explanations * **Problem Solved:** Instead of explaining individual predictions post-hoc, this family aims to design GNNs that are inherently interpretable or provide global insights into the model’s general decision-making logic. This addresses the need for model-level transparency. * **Core Innovation & Mechanism:** These approaches embed interpretability directly into the GNN architecture or learn general patterns that characterize the model’s behavior. **XGNN (19):** *XGNN focuses on model-level* explanations.* It learns a policy network that generates graph structures (e.g., motifs, patterns) that are most likely to trigger a specific prediction from the target GNN. This allows users to understand what types of graph patterns the GNN generally responds to, providing a global perspective on its learned logic. * **ProtGNN (54):** ProtGNN is a prototype-based GNN that learns a set of representative prototypes (subgraphs) during training. When making a prediction, it explains the decision by identifying which prototypes are most similar to the input graph, providing a human-understandable "reason by example." This is similar to case-based reasoning. * **KerGNNs (215):** Interpretable GNNs with Graph Kernels (KerGNNs) leverage the interpretability of graph kernels. By mapping graphs into a feature space where similarity is easily understood, KerGNNs aim to make the GNN’s decision process more transparent through these kernel-based representations. * **Invariant**

Rationales (64): This work focuses on discovering invariant rationales for GNNs, aiming to identify stable and robust explanatory patterns that hold across different contexts or perturbations, contributing to more reliable explanations. * **Global Interactive Patterns (67):** This research aims to unveil global interactive patterns across graphs, moving towards interpretable GNNs by identifying how different parts of the graph interact to influence overall predictions. * **Evidence:** XGNN (19) provides insights into the types of graph structures that influence a GNN’s decisions, which can be crucial for understanding model biases or vulnerabilities. ProtGNN (54) offers human-understandable prototype explanations, making the reasoning process transparent through concrete examples. * **Theoretical Limitations:** Intrinsically interpretable models often face a trade-off with model expressiveness and accuracy. Simpler, more transparent architectures might not capture the full complexity of real-world graph data as effectively as complex black-box GNNs. The definition of "global insight" can also be abstract and hard to quantify. * **Practical Limitations:** Learning prototypes or generating complex graph patterns (as in XGNN) can be computationally resource-intensive. The generalizability of global explanations to specific, nuanced local predictions might be limited. ***Comparison:** This family contrasts sharply with post-hoc methods. While post-hoc methods (e.g., SubgraphX (8)) explain a specific prediction, intrinsically interpretable models (e.g., ProtGNN (54)) or global explainers (e.g., XGNN (19)) aim to explain the model itself or its general behavior. XGNN (19) provides model-level explanations by generating patterns, which is distinct from instance-level* explanations.*

Synthesis and Implications: The field of XGNNs is characterized by a fundamental tension between the fidelity of explanations (how accurately they reflect the model’s true reasoning) and their interpretability (how easily humans can understand them). The evolution shows a clear trend from simple feature importance to more sophisticated structural (subgraph) and counterfactual explanations, and from instance-level insights to model-level understanding. A major unresolved debate is the lack of standardized, objective metrics for evaluating the "goodness" of explanations (43), making direct comparisons challenging. Most papers implicitly assume that providing explanations will lead to in-

creased trust, but the cognitive aspects of human-AI interaction with XGNNs are still underexplored. Future work needs to focus on more robust and generalizable explanation methods, better evaluation protocols, and potentially hybrid approaches that combine the strengths of both local and global explanations.

Fairness and Bias Mitigation in GNNs

The increasing deployment of GNNs in critical decision-making processes, such as loan default prediction (272), credit risk assessment (297), and recommender systems (4), necessitates a rigorous examination of their fairness. GNNs, by their very nature of propagating information across interconnected entities, are particularly susceptible to learning and amplifying biases present in the training data, leading to discriminatory outcomes. This section explores the crucial research dedicated to ensuring fairness and mitigating bias in GNNs, building upon the broader discussion of data imperfections and robustness from Section 4. The goal is to ensure that GNNs operate equitably, preventing disparate impacts that can arise from training data disparities or structural inequalities in the graph.

Context and Motivation: Bias in GNNs can manifest in several ways: (1) **Node feature bias:** Disparities in attributes (e.g., sensitive demographic information) can lead to biased representations. (2) **Structural bias:** Graph topology itself might reflect societal biases (e.g., homophily leading to segregation, or certain groups having fewer connections). (3) **Algorithmic bias:** The GNN’s learning process might inadvertently amplify these biases during message passing. The consequences of unfair GNNs can be severe, ranging from perpetuating social inequalities to eroding public trust. Therefore, developing techniques to identify, measure, and mitigate these biases is paramount for the ethical and responsible deployment of GNNs. A key challenge is the trade-off between achieving fairness and maintaining predictive utility, as debiasing often comes with a performance cost.

Method Family A: Pre-processing and In-processing Debiasing * Problem Solved: This family aims to mitigate biases in GNN predictions by either modifying the

input data before training (pre-processing) or incorporating fairness-aware mechanisms directly into the GNN’s training process (in-processing). This addresses the root causes of bias by intervening early in the model lifecycle. * **Core Innovation & Mechanism:**

- * **Pre-processing: Data Debiasing and Graph Rewiring:** * **EDITS (11):** This method focuses on modeling and mitigating data bias for GNNs. It identifies and corrects biased information propagation by analyzing the underlying data distribution and adjusting it to reduce disparities before the GNN processes it. This can involve re-weighting nodes or edges, or modifying node features to be less correlated with sensitive attributes.
- * **Graph Rewiring for Fairness:** Similar to robustness efforts in Section 4 (257), graph rewiring can be used to create a more balanced or less biased graph structure. This might involve adding connections between underrepresented groups or removing spurious links that perpetuate bias.
- * **In-processing: Fairness-Aware GNN Training:** These methods integrate fairness considerations directly into the GNN’s learning objective or architecture.
- * **Fair GNNs with Limited Sensitive Attribute Information (98):** This work addresses the practical challenge where sensitive attribute information might be limited or unavailable. It proposes methods to learn fair GNNs by inferring or approximating sensitive attributes, or by using proxy features, and then applying debiasing techniques.
- * **Individual Fairness for GNNs (9):** *This paper focuses on individual fairness**, ensuring that similar individuals (nodes) receive similar predictions, regardless of their sensitive attributes. It proposes a ranking-based approach to achieve this, aiming for consistency in outcomes for comparable entities.
- * **Debiasing via Learning Disentangled Causal Substructure (25):** This innovative approach aims to debias GNNs by learning disentangled causal substructures. The core idea is to separate features that causally influence the outcome from those that are merely correlated with sensitive attributes (confounders). By disentangling these factors, the GNN can make predictions based on causal relationships, reducing spurious bias.
- * **Rethinking Fair GNNs from Re-balancing (12):** This work re-examines fairness through a re-balancing lens, proposing techniques that adjust the weights of samples or the aggregation process during training to ensure that different groups are treated equitably. This can involve re-weighting

loss contributions or modifying message passing to counteract biased information flow. *

Mitigating Sensitive Attribute Leakage (116): This research focuses on preventing GNNs from inferring sensitive attributes from non-sensitive features or graph structure and then using this leaked information to make biased predictions. It proposes mechanisms to obscure or decorrelate sensitive information during representation learning. *

Evidence: Pre-processing methods like EDITS (11) have shown effectiveness in reducing bias metrics while maintaining competitive utility. In-processing methods, such as those by (25) and (12), demonstrate significant improvements in various fairness metrics (e.g., demographic parity, equal opportunity) on benchmark datasets, often with a controlled trade-off in accuracy. (116) shows that mitigating attribute leakage can lead to fairer outcomes. *

Theoretical Limitations: Defining and measuring fairness in complex graph contexts is inherently challenging, with multiple, sometimes conflicting, definitions (e.g., individual vs. group fairness, disparate treatment vs. disparate impact). The underlying causal mechanisms of bias in graphs are often difficult to model explicitly, making disentanglement complex (25). The trade-off between fairness and utility is a persistent theoretical tension (322). *

Practical Limitations: Many methods require access to sensitive attribute information, which might be unavailable, incomplete, or legally restricted (e.g., due to privacy concerns). Implementing sophisticated debiasing techniques can add computational overhead and complexity to GNN training. The generalizability of debiasing methods across different graph types and bias definitions is also an open question. *

Comparison: Pre-processing methods (e.g., EDITS (11)) are generally simpler to implement but might not fully address biases introduced during the GNN’s learning process. In-processing methods (e.g., (25; 12; 116)) offer more fine-grained control over bias mitigation by integrating fairness directly into the model’s learning, but are often more complex. The evolution shows a clear trend towards more sophisticated in-processing methods that aim to address bias at a deeper, causal level.

Method Family B: Post-processing Debiasing *Problem Solved:* *Adjusting the predictions of a trained GNN after* inference to achieve fairer outcomes, without modifying the model itself. This addresses scenarios where model re-training is not fea-*

sible or sensitive attributes are only available post-inference. * **Core Innovation & Mechanism:** These methods typically involve re-calibrating prediction scores or adjusting decision thresholds based on fairness criteria. For example, if a model shows disparate impact for a certain group, its predictions for that group might be adjusted to align with a desired fairness metric (e.g., equalizing false positive rates across groups). * **Evidence:** Can be effective in simple, well-defined scenarios to achieve specific fairness metrics. * **Theoretical Limitations:** Post-processing cannot correct fundamental biases learned by the GNN during training; it merely adjusts the output. This means the underlying biased reasoning of the model remains. It also often requires access to sensitive attributes to perform the adjustment. * **Practical Limitations:** The adjustments can sometimes be non-transparent, obscuring the original model’s reasoning. It might not generalize well across different fairness metrics or complex decision landscapes. * **Comparison:** Post-processing is generally the least invasive but also the least powerful debiasing strategy. It’s a reactive measure, contrasting with the proactive nature of pre-processing and in-processing methods that aim to prevent bias from being learned in the first place.

Synthesis and Implications: The field of fairness in GNNs is rapidly evolving, moving from simple demographic parity adjustments to more nuanced approaches that consider individual fairness (9), causal disentanglement (25), and sensitive attribute leakage (116). A critical tension exists between achieving different fairness notions, as optimizing for one (e.g., equal opportunity) might degrade another (e.g., demographic parity). The fundamental trade-off between fairness and accuracy remains a central challenge, with researchers continually seeking methods that minimize this conflict. Most research implicitly assumes that sensitive attributes are well-defined, but in reality, these can be fluid and context-dependent. Future research needs to focus on robust causal inference in graphs, developing methods that work with limited or no sensitive attribute information, and creating unified frameworks that can navigate the complex interplay between different fairness definitions and other trustworthiness dimensions like explainability and privacy.

Privacy-Preserving Graph Neural Networks

The interconnected nature of graph data, which often represents relationships between individuals, organizations, or sensitive entities, makes privacy a paramount concern for Graph Neural Networks (GNNs). Deploying GNNs in domains like social networks, medical records, financial transactions, or even cybersecurity (21; 191) risks exposing sensitive information about nodes (individuals) or edges (relationships). Adversaries might attempt to infer private attributes, reconstruct parts of the graph, or even identify individuals from the GNN model or its outputs. This section examines the critical efforts to enhance privacy in GNNs, including methods to protect sensitive information during training and inference, thereby fostering greater confidence and ethical use of GNN technology. This area builds upon the general architectural principles of GNNs (Section 2) and is closely related to the robustness against adversarial attacks (Section 4), as some privacy attacks are essentially inference attacks.

Context and Motivation: The message-passing paradigm, which aggregates information from neighbors, is a double-edged sword: while it enables powerful representation learning, it also facilitates the propagation and potential leakage of sensitive information. For example, a GNN trained on a social network might inadvertently reveal a user’s political affiliation or health status based on their connections and interactions. Traditional data anonymization techniques often fall short for graphs, as structural patterns can still enable re-identification. Therefore, specialized privacy-preserving techniques are essential to safeguard sensitive information, comply with regulations (e.g., GDPR, HIPAA), and maintain user trust. The central challenge often revolves around balancing strong privacy guarantees with acceptable model utility and computational efficiency.

Method Family A: Differential Privacy (DP) for GNNs * **Problem Solved:** Differential Privacy (DP) provides a strong, mathematically rigorous guarantee that the presence or absence of any single individual’s data in the training set does not significantly alter the model’s output. For GNNs, this means protecting individual node features or edges from being inferred from the trained model. * **Core Innovation & Mechanism:** DP is typically achieved by injecting calibrated noise into the learning process. * **DP-**

SGD for GNNs: The most common approach involves adapting Differentially Private Stochastic Gradient Descent (DP-SGD) to GNN training. During each training step, noise is added to the gradients computed for each mini-batch before they are used to update the model parameters. This ensures that the gradient contribution of any single data point is obscured. * **Output Perturbation:** Alternatively, noise can be added to the final node embeddings or predictions generated by the GNN. This provides privacy for the model’s outputs but does not protect the training process itself. * **Evidence:** DP-GNNs offer quantifiable privacy guarantees, expressed by a privacy budget (ϵ). A smaller ϵ implies stronger privacy. While specific quantitative results vary, DP has been shown to provide strong protection against various inference attacks, including membership inference and attribute inference. * **Theoretical Limitations:** DP often introduces a significant trade-off with model utility (accuracy). Stronger privacy guarantees (smaller ϵ) typically lead to a more substantial degradation in performance. The optimal amount and type of noise to add, especially for graph structures, is an active research area. Applying DP to graph topology (edges) is more complex than to node features, as small changes in structure can have a large impact on graph properties. * **Practical Limitations:** DP-SGD can be computationally more expensive than standard SGD due to the per-example gradient clipping and noise addition. Tuning the privacy parameters (e.g., ϵ , noise scale) requires expertise and careful consideration of the privacy-utility trade-off. The utility degradation can make DP-GNNs less competitive with non-private counterparts in some tasks. * **Comparison:** DP offers the strongest theoretical privacy guarantees among the methods discussed, making it a gold standard for privacy. However, this often comes at a higher cost in terms of model utility and computational overhead compared to federated learning, which focuses on data locality rather than noise injection.

Method Family B: Federated Learning (FL) for GNNs * **Problem Solved:** Federated Learning enables the collaborative training of a global GNN model across multiple decentralized data sources (clients) without requiring clients to share their raw graph data. This protects data locality and individual privacy by keeping sensitive information on local devices. * **Core Innovation & Mechanism:** The FL paradigm is adapted for

graph-structured data. * **FedGraphNN (185)**: This work proposes a comprehensive federated learning system and benchmark specifically for GNNs. It addresses challenges unique to FL on graphs, such as data heterogeneity (clients having different graph structures or feature distributions), communication efficiency (reducing the amount of data exchanged between clients and server), and the aggregation of graph-specific model updates. Clients train local GNNs on their private subgraphs or full graphs, and only model parameters or gradients are sent to a central server for aggregation. * **Graph-specific FL Challenges (195)**: A survey by (195) highlights several challenges: (1) **Non-IID data**: Graph data is often inherently non-IID across clients, which can degrade FL performance. (2) **Graph partitioning**: How to partition a large graph across clients while preserving useful local structure is crucial. (3) **Communication overhead**: Aggregating GNN models can still be substantial. (4) **Privacy attacks**: FL alone is not immune to inference attacks (e.g., reconstructing data from shared gradients) or poisoning attacks by malicious clients, often necessitating additional privacy mechanisms like DP or secure aggregation. * **Evidence**: FL-GNNs enable collaborative model training in scenarios where data cannot be centrally collected due to privacy regulations or logistical constraints (e.g., training a GNN for disease prediction across multiple hospitals, or for traffic prediction across different city departments). * **Theoretical Limitations**: While FL prevents direct sharing of raw data, it does not provide the same strong, provable privacy guarantees as DP against sophisticated inference attacks on gradients or model updates. The effectiveness of FL is highly dependent on the degree of data heterogeneity across clients. * **Practical Limitations**: Communication overhead can still be significant for large GNN models or frequent updates. Managing and synchronizing training across many clients can be complex. The system is vulnerable to malicious clients who might send poisoned updates or attempt to infer information from aggregated models. * **Comparison**: FL provides a practical approach to privacy by decentralizing data, contrasting with DP’s noise injection. It is particularly relevant for scenarios where data cannot be centrally collected. However, FL often needs to be combined with other PETs (like DP) to achieve stronger privacy guarantees against advanced attacks.

Method Family C: Other Privacy-Enhancing Technologies (PETs) and Anonymization

*** Problem Solved:** Protecting specific sensitive information (e.g., link existence, node identity, attribute values) through cryptographic methods or data transformation, often providing very strong privacy guarantees.

*** Core Innovation & Mechanism:**

*** Homomorphic Encryption (HE):** HE allows computations to be performed on encrypted data without decrypting it. A GNN could theoretically operate on encrypted node features and adjacency matrices, producing encrypted predictions. This means a server could train or infer with a GNN without ever seeing the raw sensitive data.

*** Secure Multi-Party Computation (SMC):** SMC enables multiple parties to jointly compute a function over their private inputs while ensuring that no party learns anything about the other parties' inputs beyond what can be inferred from the function's output. This could be used for private aggregation of gradients or features in a GNN.

*** Graph Anonymization:** This involves modifying the graph structure or node attributes to prevent re-identification of individuals or sensitive links. Techniques include k-anonymity (ensuring each node is indistinguishable from at least k-1 other nodes), edge generalization, or adding/removing edges to obscure sensitive patterns.

*** Evidence:** HE and SMC offer cryptographic-level privacy guarantees, making them highly secure against various attacks. Graph anonymization can be effective in preventing re-identification attacks.

*** Theoretical Limitations:** HE and SMC are notoriously computationally expensive, often increasing computation time by several orders of magnitude, making them currently impractical for training large-scale GNNs. Graph anonymization, while effective for re-identification, often significantly distorts the original graph properties, leading to a substantial degradation in model utility and potentially altering the underlying graph semantics.

*** Practical Limitations:** The high computational cost and complexity of implementing HE and SMC are major barriers to their widespread adoption for GNNs. Graph anonymization techniques require careful tuning to balance privacy and utility, and the choice of anonymization strategy can heavily influence the downstream GNN performance.

*** Comparison:** These methods offer the strongest privacy guarantees, often cryptographic, but come with the highest computational and implementation costs.

They represent the "gold standard" of privacy but are far from scalable for current GNNs. They contrast with DP and FL by offering a different paradigm for privacy, focusing on cryptographic protection or structural obfuscation rather than noise injection or data decentralization.

Synthesis and Implications: The field of privacy-preserving GNNs is characterized by a fundamental and persistent trade-off between privacy guarantees and model utility/efficiency. The evolution shows a spectrum of solutions, from strong theoretical guarantees of DP, through the practical distributed training of FL, to the high-security but computationally intensive cryptographic methods. A major unresolved challenge is developing scalable and efficient privacy-preserving GNNs that can achieve strong privacy without severely compromising performance. Most research implicitly assumes that the adversary model is well-defined, but real-world attacks can be more sophisticated. Future research needs to focus on hybrid approaches that combine the strengths of different PETs (e.g., FL with DP or secure aggregation), developing more efficient cryptographic schemes, and creating robust privacy-preserving methods that can handle the unique complexities of dynamic and heterogeneous graph data. The interplay between privacy, fairness, and explainability is also a critical area for future investigation, as optimizing for one dimension might inadvertently impact the others.

6. Key Challenges and Open Problems

Despite the remarkable advancements in Graph Neural Networks (GNNs) over the past decade, transforming their application across diverse fields from drug discovery (165; 188; 274; 275; 277) to recommender systems (111; 79; 89; 92; 35; 155; 182; 223; 234; 259), significant challenges persist that hinder their full potential and reliable deployment in real-world, complex scenarios (146; 42; 235). While previous sections have explored the architectural foundations (Section 2), expressive power (Section 3), robustness against adversarial attacks (Section 4), and the critical need for trustworthiness (Section 5), these advancements often come with their own set of limitations or introduce new complexities. This section consolidates and critically analyzes these persistent challenges and

open problems, highlighting the intellectual frontiers that demand further research and innovation.

The field is currently grappling with a fundamental tension between the theoretical elegance of GNNs and the practical realities of their application. Early GNN models, primarily designed for static, relatively small graphs with strong homophily, are increasingly inadequate for the massive, dynamic, and often heterophilous graphs encountered in real-world systems (? 108). This has led to a paradigm shift, where researchers are moving beyond incremental architectural tweaks to address systemic bottlenecks. One of the most pressing issues is **scalability**, as the message-passing paradigm, while powerful, becomes computationally prohibitive for graphs with billions of nodes and edges, and struggles to adapt to constantly evolving graph structures (198; 33). This challenge is exacerbated by the desire for **deeper and more expressive GNNs**, which often introduce or amplify problems like over-smoothing (where node representations become indistinguishable) and over-squashing (where information from distant nodes is lost), thereby limiting their ability to capture long-range dependencies and complex structural patterns (133; 81; 84; 59).

Furthermore, the performance of GNNs often degrades significantly when deployed on **unseen graph structures or under distribution shifts**, raising critical questions about their generalization capabilities. Unlike traditional deep learning models that generalize well to new samples from the same distribution, GNNs must generalize to entirely new *graph topologies* and feature distributions, a far more challenging task (228). This inductive generalization problem is compounded by the inherent biases and noise often present in real-world graph data, which GNNs can inadvertently learn and perpetuate. Finally, the rapid proliferation of GNN architectures and applications has outpaced the development of **robust evaluation methodologies and standardized benchmarking frameworks**. This lack of consistent and realistic evaluation practices makes it difficult to objectively compare models, identify true progress, and ensure reproducibility, potentially leading to misleading performance claims and fragmented research efforts (5; 44). Addressing these interconnected challenges is crucial for GNNs to transition from powerful research tools to reliable and trustworthy components of real-world intelligent

systems.

Scalability to Large and Dynamic Graphs

The success of Graph Neural Networks in various domains has highlighted a critical bottleneck: their inherent difficulty in scaling to massive, real-world graphs and effectively handling dynamic graph structures that evolve over time (121; 123). While early GNNs demonstrated impressive performance on relatively small, static datasets, real-world graphs in social networks, recommender systems (4; 92), biological systems (167; 178; 158), and transportation networks (130; 110; 170) often comprise billions of nodes and edges, and their topologies and features are constantly changing. This challenge is multifaceted, encompassing computational complexity, memory constraints, and the need for adaptive learning mechanisms.

Context and Motivation: The core message-passing paradigm of GNNs, where node representations are iteratively updated by aggregating information from their local neighborhoods, leads to an exponential growth in the computational graph size with increasing depth. For a GNN with L layers, each node’s representation depends on its L -hop neighborhood. In large graphs, these neighborhoods can become prohibitively vast, leading to excessive memory consumption and slow training times (129). Furthermore, real-world graphs are rarely static; new nodes and edges appear, existing ones disappear, and features change. Traditional GNNs, designed for static graphs, require expensive re-training or complex incremental updates to adapt to such dynamism, a problem addressed by works like (33) and (110).

Method Family A: Sampling-based and Distributed Approaches for Static Large Graphs * **Problem Solved:** These methods aim to reduce the computational and memory footprint of GNNs on large static graphs, making training feasible. They address the practical limitations of full-batch training on massive datasets. * **Core Innovation & Mechanism:** * **Sampling-based Methods:** Approaches like GraphSAGE (?) and Cluster-GCN (?) tackle scalability by sampling a subset of neighbors for each node during message passing or by partitioning the graph into subgraphs. Graph-

SAGE, for instance, samples a fixed number of neighbors at each layer, effectively creating a fixed-size computational graph for each node. This significantly reduces the memory footprint and computation per node. * **Distributed Training Frameworks:** For extremely large graphs that cannot fit into a single machine’s memory, distributed training frameworks like DistGNN (198) and the Deep Graph Library (DGL) (129) partition the graph and distribute computation across multiple machines. These systems manage data partitioning, communication, and synchronization of model parameters. * **Conditions for Success:** Sampling methods succeed when the sampled neighborhoods are representative enough to preserve crucial information for learning. Distributed systems require robust communication protocols and efficient graph partitioning strategies. * **Theoretical Limitations:** Sampling introduces variance into gradient estimates, potentially slowing down convergence or leading to suboptimal solutions. The choice of sampling strategy can significantly impact the quality of learned representations. Distributed training faces challenges with communication overhead and maintaining data consistency across nodes. * **Practical Limitations:** Sampling can lead to a loss of global context, as information from distant, unsampled nodes is ignored. Distributed systems are complex to set up and manage, requiring significant engineering effort. The communication costs can become a bottleneck, especially for dense graphs or models with many parameters. ***Comparison:** Sampling methods (e.g., GraphSAGE) reduce the per-node computation but might lose global context. Distributed frameworks (e.g., DistGNN (198)) aim to scale the entire graph* computation by parallelization. A critical tension exists between reducing computational complexity (via sampling) and preserving the integrity of graph information. While sampling offers a simpler, single-machine solution, distributed training is necessary for truly massive graphs, albeit with increased infrastructure complexity.*

Method Family B: Non-Convolutional and Decoupled Approaches for Scalability * **Problem Solved:** These methods offer alternative paradigms to the traditional message-passing framework, aiming to inherently improve scalability by decoupling computation or by adopting fundamentally different aggregation mechanisms. * **Core Innovation & Mechanism:** * **Decoupled GNNs:** Approaches like SIGN (63) and

those that decouple feature transformation from propagation (e.g., (18)) pre-compute aggregated features for multiple hops and then apply a simple MLP. This avoids iterative message passing during training, making it highly efficient.

* **Non-Convolutional GNNs:** The Random Walk with Unifying Memory (RUM) neural network (1) represents a significant departure by being entirely convolution-free. It stochastically samples finite-length random walks for each node and processes these walk trajectories using an RNN to form node representations. The key innovation is its runtime complexity of $O(|V|lkD)$, which is agnostic to the number of edges $|E|$, making it particularly efficient for dense graphs. RUM naturally supports mini-batching and is scalable without requiring all neighbors to be present.

Global Aggregation with Linear Complexity: GloGNN (2) addresses the challenge of heterophily (discussed further in 6.3) but also introduces an acceleration technique that reduces its global aggregation time complexity from cubic/quadratic to linear* ($O(k^2n)$). This is achieved by leveraging a learned coefficient matrix and reordering matrix multiplications, making global information aggregation feasible for larger graphs.

* **Conditions for Success:** Decoupled GNNs work well when the pre-computed features are sufficiently informative. RUM (1) relies on the ability of random walks to capture relevant structural and semantic information. GloGNN (2) succeeds by efficiently learning global correlations.

* **Theoretical Limitations:** Decoupled GNNs might lose some of the dynamic interaction benefits of iterative message passing. RUM’s theoretical guarantees for expressiveness and over-smoothing alleviation rely on assumptions about universal and injective functions (1). GloGNN’s linear complexity depends on the number of labels (c) and the sparsity of the adjacency matrix.

* **Practical Limitations:** Decoupled methods can still suffer from over-smoothing if the propagation depth is too high. RUM’s performance can depend on the length and number of random walks sampled. While GloGNN achieves linear complexity, the constant factors and memory usage for the coefficient matrix might still be considerable for extremely large graphs.

* **Comparison:** Decoupled GNNs (e.g., SIGN (63)) offer speed by pre-computation but might sacrifice some representational power. RUM (1) offers a fundamentally different, non-convolutional approach that is theoretically more expressive

and empirically faster than even simple convolutional GNNs (Figure 4 in (1)), while also addressing over-smoothing and over-squashing. GloGNN (2) provides an efficient way to aggregate global information, which is a common bottleneck for many GNNs. This evolution shows a clear shift from simply reducing graph size to rethinking the core aggregation mechanism for scalability.

Method Family C: Dynamic Graph Neural Networks (DGNNs) * **Problem Solved:** DGNNs specifically address the challenge of graphs that evolve over time, where nodes, edges, or their attributes change. They aim to efficiently update node representations and predictions without full re-training. * **Core Innovation & Mechanism:** * **Continuous-Time DGNNs:** Models surveyed by (33) and (110) for time series and urban computing, respectively, explicitly model the temporal evolution of graphs. They often use recurrent neural networks (RNNs) or attention mechanisms to process sequences of graph snapshots or continuous-time events. For example, (145) proposes Dynamic Multiscale GNNs for 3D skeleton-based human motion prediction, capturing fine-grained temporal dynamics. * **Incremental Updates:** Instead of re-training, DGNNs often focus on incrementally updating node embeddings based on new events (e.g., edge additions/deletions). This is crucial for real-time applications. * **Spatio-Temporal GNNs:** For applications like traffic forecasting (130; 126; 263) or weather prediction (144; 294), GNNs are extended to spatio-temporal settings, integrating both spatial graph dependencies and temporal sequences. * **Conditions for Success:** DGNNs require efficient mechanisms to store and retrieve historical information, and robust methods to integrate new events into existing representations. * **Theoretical Limitations:** Modeling complex temporal dependencies in graphs is challenging. The definition of "dynamic" can vary (discrete snapshots vs. continuous events), leading to diverse model complexities. The interplay between spatial and temporal information can be intricate. * **Practical Limitations:** Storing and processing historical graph data can be memory-intensive. The computational cost of updating representations for every new event can still be high, especially in high-throughput systems. The lack of standardized dynamic graph benchmarks makes comparisons difficult. * **Comparison:** DGNNs are distinct from static

graph scalability solutions as they explicitly handle the temporal dimension. While static methods focus on reducing the snapshot size, DGNNs focus on efficiently evolving representations. The survey by (33) highlights the state-of-the-art and open challenges in this area, emphasizing the need for more robust and scalable solutions.

Synthesis and Implications: The scalability of GNNs remains a formidable challenge, with solutions diverging along several dimensions: reducing the effective graph size (sampling), parallelizing computation (distributed), fundamentally altering the aggregation mechanism (non-convolutional), or explicitly modeling temporal evolution (dynamic GNNs). A critical tension exists between achieving high scalability and preserving the rich structural information of graphs. While methods like RUM (1) and GloGNN (2) offer promising avenues by rethinking the core GNN operations, the problem of truly massive, continuously evolving graphs still requires more generalizable and efficient solutions. Future research needs to focus on hybrid approaches that combine the strengths of these paradigms, developing more efficient memory management for dynamic graphs, and exploring hardware-aware GNN designs.

Persistent Issues with Expressiveness and Over-squashing

Despite significant architectural innovations, Graph Neural Networks continue to grapple with fundamental limitations in their expressive power and the pervasive problems of over-smoothing and over-squashing (118). As discussed in Section 3, many standard GNNs are theoretically no more powerful than the 1-dimensional Weisfeiler-Leman (1-WL) graph isomorphism test (122; 65), limiting their ability to distinguish between structurally different but 1-WL equivalent graphs. This inherent limitation prevents them from capturing complex structural patterns, such as cycles or specific motifs, which are crucial for many real-world tasks. Furthermore, the quest for deeper GNNs, often motivated by the success of deep CNNs, has exacerbated the issues of over-smoothing and over-squashing, hindering the learning of long-range dependencies (133; 81; 84; 59).

Context and Motivation: The expressive power of a GNN dictates the range of graph structures and properties it can learn and distinguish. If a GNN cannot differentiate

between two non-isomorphic graphs, it will necessarily make the same prediction for both, regardless of their true labels. This is a critical theoretical bottleneck. Over-smoothing, where node representations converge to similar values across the graph with increasing layers, leads to a loss of discriminative power for individual nodes (59). Over-squashing, on the other hand, refers to the information bottleneck created when a large k -hop neighborhood’s information must be compressed into a fixed-size vector at the central node, making it difficult for gradients to flow effectively from distant nodes and thus hindering the capture of long-range dependencies (84). These issues are particularly salient in applications requiring fine-grained structural understanding, such as molecular property prediction (99; 51; 175; 169; 244; 287; 253; 254) or social network analysis.

Method Family A: Enhancing Expressiveness Beyond 1-WL * **Problem Solved:** This family of methods aims to overcome the 1-WL expressiveness bottleneck, enabling GNNs to distinguish more complex graph structures and capture richer topological information. * **Core Innovation & Mechanism:** * **Higher-Order GNNs:** Approaches like k -WL GNNs (65) or those based on higher-order graph convolutions explicitly consider subgraphs or tuples of nodes. They effectively simulate higher-dimensional WL tests, which are known to be more powerful. For example, (60) investigates whether GNNs can count substructures, a task that often requires higher expressive power. * **Path Neural Networks (PathNNs):** (6) introduces PathNNs, which update node representations by aggregating information from various paths emanating from each node. Their core innovation lies in operating on "annotated sets of paths," where nodes within paths are recursively annotated with hashes of shorter path sets. This allows PathNNs to be strictly more powerful than 1-WL and, for the most expressive variant ($\tilde{A}P$), even distinguish graphs indistinguishable by the 3-WL algorithm, empirically validated on synthetic datasets. * **Non-Convolutional GNNs:** The Random Walk with Unifying Memory (RUM) neural network (1) is theoretically shown to be more expressive than the 1-WL test, capable of distinguishing non-isomorphic graphs that WL-equivalent GNNs cannot (e.g., cycle sizes, radius). Its innovation stems from processing random walk trajectories with an RNN that merges semantic and topological features, moving away from local

message passing. * **Geometric GNNs:** Works like (14) and (102; 124; 226) explore the expressive power of geometric GNNs, which incorporate geometric information (e.g., node coordinates, distances) to enhance their ability to distinguish structures. * **Conditions for Success:** Higher-order GNNs require careful design to manage exponential complexity. PathNNs (6) succeed by efficiently encoding and aggregating path information up to a fixed length. RUM (1) relies on the richness of random walk trajectories. * **Theoretical Limitations:** Higher-order GNNs face exponential computational complexity, making them impractical for large graphs. PathNNs’ AP variant is NP-hard for finding all simple paths, necessitating length constraints (6). RUM’s theoretical proofs rely on assumptions about universal and injective functions (1). * **Practical Limitations:** The increased expressiveness often comes at a significant computational cost, limiting the depth or size of graphs that can be processed. Implementing and optimizing these complex architectures can be challenging. * **Comparison:** PathNNs (6) offer a distinct path-centric approach to expressiveness, directly leveraging structural context beyond immediate neighbors, and showing superior power to 1-WL and even 3-WL. This contrasts with higher-order GNNs that explicitly consider larger subgraphs. RUM (1) provides a non-convolutional alternative that also surpasses 1-WL, offering a different mechanism for capturing complex structures. The evolution highlights a move towards more sophisticated structural encoding and non-local information aggregation.

Method Family B: Mitigating Over-smoothing and Over-squashing * **Problem Solved:** These methods aim to prevent node representations from becoming indistinguishable (over-smoothing) and to facilitate the flow of information and gradients over long distances (over-squashing). * **Core Innovation & Mechanism:** * **Residual and Skip Connections:** Inspired by deep learning, architectures like GCNII (?) and DeepGCNs (?) incorporate residual connections or initial residual connections to preserve the initial node features, allowing for deeper GNNs by mitigating over-smoothing. (32) also explores strategies for deeper GNNs. * **Adaptive Propagation and Decoupling:** Methods like GPR-GNN (114) and those that decouple feature transformation from propagation (18) allow for more flexible message passing, often with learnable weights that adapt to lo-

cal graph properties. This can prevent excessive smoothing by controlling the influence of neighbors.

- * **Fractional Calculus:** FROND (3) introduces a novel approach by replacing integer-order differential operators in continuous GNNs with Caputo fractional derivatives. This allows FROND to inherently integrate the entire historical trajectory of node features, enabling memory-dependent dynamics. Theoretically, its non-Markovian random walk interpretation leads to a slow algebraic rate of convergence to stationarity, which inherently mitigates over-smoothing, unlike the exponential convergence in Markovian integer-order models.
- * **Hierarchical Pooling and Global Attention:** To address over-squashing, hierarchical GNNs (e.g., DiffPool (?), SAGPool (?), Graph U-Nets (?), surveyed in (29)) coarsen the graph, aggregating information at different scales. Graph Transformers (7) and global attention mechanisms (135) directly connect distant nodes, bypassing the local message-passing bottleneck and allowing for long-range interactions.
- * **Non-Convolutional Approaches:** RUM (1) is also shown to alleviate over-smoothing (its expected Dirichlet energy does not diminish with long walks) and over-squashing (decays slower in inter-node Jacobian). This is attributed to its non-convolutional nature and the way it processes walk trajectories.
- * **Conditions for Success:** Residual connections are effective for moderately deep GNNs. FROND (3) is particularly beneficial for graphs exhibiting non-local, memory-dependent behaviors. Pooling methods require effective coarsening strategies. Global attention is powerful but computationally intensive.
- * **Theoretical Limitations:** While residual connections help, they don't fundamentally change the message-passing paradigm, and over-smoothing can still occur in very deep models (59). Fractional calculus in FROND (3) relies on numerical FDE solvers, which can have computational considerations. Global attention mechanisms often incur quadratic complexity, limiting scalability.
- * **Practical Limitations:** Deep GNNs, even with skip connections, can be hard to train. Pooling layers introduce information loss and require careful design. Global attention is memory-intensive for large graphs. The choice of optimal fractional order in FROND (3) may require domain knowledge or extensive tuning.
- * **Comparison:** Residual connections are a simple, widely adopted technique, but FROND (3) offers a more principled, theoretically grounded approach to mitigating

over-smoothing by fundamentally altering the dynamics of feature evolution. Pooling and global attention are distinct strategies for tackling over-squashing, with pooling offering hierarchical abstraction and attention providing direct long-range connections. RUM (1) offers a unified solution to both over-smoothing and over-squashing through its non-convolutional design. The evolution shows a trend from simple architectural fixes to more fundamental re-evaluations of the GNN propagation mechanism.

Synthesis and Implications: The challenges of expressiveness, over-smoothing, and over-squashing are deeply intertwined, representing fundamental limitations of the standard message-passing paradigm. The field is actively exploring diverse solutions, ranging from higher-order and path-based GNNs (e.g., (6)) to entirely non-convolutional architectures (e.g., (1)) to enhance expressive power. Simultaneously, innovative techniques like fractional calculus in FROND (3) and hierarchical pooling are being developed to mitigate over-smoothing and over-squashing. A critical tension exists between achieving high expressiveness and maintaining computational efficiency and scalability. The unresolved debate centers on finding architectures that can simultaneously achieve high expressive power, capture long-range dependencies, and scale to large graphs without succumbing to over-smoothing. Future research needs to focus on hybrid models that combine the strengths of different approaches, developing new theoretical frameworks for understanding information flow in deep GNNs, and exploring adaptive mechanisms that can dynamically adjust propagation based on local graph properties.

Generalization to Unseen Structures and Distribution Shifts

A significant open problem for Graph Neural Networks is their ability to generalize effectively to unseen graph structures and mitigate performance drops due to distribution shifts (228). Unlike standard deep learning models that operate on i.i.d. data, GNNs learn from graph-structured data, where the underlying topology itself can vary significantly between training and test sets. This inductive generalization challenge is paramount for deploying GNNs in dynamic, open-world environments where the graph structure or node/edge feature distributions are constantly evolving or differ from the training domain.

This problem is closely related to the robustness concerns discussed in Section 4, as poor generalization can be seen as a form of vulnerability to out-of-distribution inputs.

Context and Motivation: GNNs learn an inductive bias from the training graph’s structure and feature distribution. When presented with graphs that have different structural properties (e.g., different average degree, community structure, homophily levels) or feature distributions, their performance can degrade substantially. This is particularly problematic in applications like anomaly detection (142; 227; 221; 186), where anomalies often manifest as novel graph patterns, or in scientific domains like materials science (51) and chemistry (175), where new molecular graphs must be predicted. The challenge is to develop GNNs that learn fundamental, transferable principles of graph structure and function, rather than overfitting to specific training graph characteristics.

Method Family A: Pre-training and Data Augmentation for Generalization * **Problem Solved:** These methods aim to improve the generalizability of GNNs by providing them with richer, more diverse training signals, either through unsupervised pre-training on large datasets or by artificially expanding the training data. * **Core Innovation & Mechanism:** * **Pre-training GNNs:** Inspired by the success of pre-training in NLP and CV, researchers have developed strategies to pre-train GNNs on large unlabeled graphs, learning generalizable graph representations that can then be fine-tuned for downstream tasks (49). GPT-GNN (112) uses generative pre-training to learn graph structures. GPPT (15) combines graph pre-training with prompt tuning to enhance generalization. GraphPrompt (77) unifies pre-training and downstream tasks, while Mole-BERT (169) rethinks pre-training for molecular graphs. These methods learn robust feature extractors or encoders that are less sensitive to specific graph topologies. * **Data Augmentation for GNNs:** Similar to image data, graph data augmentation techniques aim to increase the diversity of training samples. DropEdge (?) randomly drops edges during training to make the GNN more robust to structural perturbations. (13) provides a broader survey of data augmentation for GNNs. GraphSMOTE (139) addresses imbalanced node classification by generating synthetic nodes and edges. * **Conditions for Success:** Pre-training is effective when large amounts of unlabeled graph

data are available and the pre-training task aligns well with downstream tasks. Data augmentation works best when the augmentations are meaningful and do not introduce excessive noise or unrealistic graph structures. * **Theoretical Limitations:** Designing effective self-supervised pre-training tasks for graphs is challenging, as the "ground truth" for structural patterns is often ambiguous. Data augmentation can sometimes introduce spurious correlations or alter the fundamental properties of the graph, potentially hindering learning. * **Practical Limitations:** Pre-training requires significant computational resources for large graphs. The transferability of pre-trained models to highly diverse downstream tasks is not always guaranteed. Data augmentation strategies often require domain expertise to design appropriate transformations. * **Comparison:** Pre-training (e.g., GPT-GNN (112), GPPT (15)) aims for broad, inductive generalization by learning rich representations from vast datasets. Data augmentation (e.g., DropEdge (?)) focuses on improving robustness to minor structural variations or noise within a specific domain. The evolution shows a trend towards leveraging large-scale unlabeled data to build more generalizable foundational models.

Method Family B: Invariant Representation Learning and Robustness to Distribution Shifts * **Problem Solved:** This family focuses on learning representations that are invariant to specific types of distribution shifts or developing GNNs that are inherently robust to noisy or imperfect input data. * **Core Innovation & Mechanism:** * **Invariant Representation Learning:** Approaches like learning disentangled causal substructures (25) or discovering invariant rationales (64) aim to extract features that are causally related to the prediction target and thus robust to changes in spurious correlations. (88) explores learning invariant representations via cluster generalization. This is a deeper form of generalization, seeking to understand the underlying causal mechanisms. * **Robustness to Noisy Graphs and Sparse Labels:** (40) focuses on robust GNNs for noisy graphs with sparse labels, where the input graph structure or labels might be imperfect. (296) proposes distribution consistency-based self-training for GNNs with sparse labels. These methods often incorporate regularization techniques or uncertainty quantification (200) to handle data imperfections. * **Adapting to Dynamic**

Distribution Shifts: (228) specifically addresses dynamic graph neural networks under spatio-temporal distribution shift, proposing mechanisms to adapt model parameters or representations as the underlying data distribution evolves over time. * **Handling Heterophily:** The challenge of generalizing to graphs with varying levels of homophily (where connected nodes are similar) or heterophily (where connected nodes are dissimilar) is a specific instance of distribution shift. GloGNN (2) addresses this by learning global homophily, while GBK-GNN (208) models both homophily and heterophily. Surveys like (72; 128; 108; 217) highlight the diverse approaches to this problem. * **Conditions for Success:** Invariant learning requires accurate identification of causal factors, which is often challenging in complex graph settings. Robustness methods succeed when the noise model is well-understood or the GNN can effectively learn to filter out noise. Adapting to dynamic shifts requires continuous monitoring and efficient update mechanisms. * **Theoretical Limitations:** Causal inference on graphs is a complex and active research area, and formalizing "invariance" in graph contexts is difficult. Robustness to noise often involves trade-offs with performance on clean data. Dynamic adaptation mechanisms can be prone to catastrophic forgetting (197) or require significant memory for historical states. * **Practical Limitations:** Implementing causal disentanglement can be complex and requires careful feature engineering. Robustness methods might not generalize to all types of noise. Continuous adaptation to distribution shifts can be computationally expensive and difficult to deploy in real-time. * **Comparison:** Invariant representation learning (e.g., (25)) is a more fundamental approach to generalization, aiming to learn stable underlying principles. Robustness to noisy data (e.g., (40)) is a practical necessity for real-world data imperfections. Adapting to dynamic shifts (e.g., (228)) is crucial for evolving environments. Addressing heterophily (e.g., (2)) is a specific structural generalization challenge. This evolution shows a move from simply improving empirical performance to developing GNNs that are theoretically more grounded in causal principles and robust to diverse real-world conditions.

Synthesis and Implications: Generalization to unseen structures and distribution shifts remains a critical hurdle for GNNs. The field is exploring diverse strategies, from

leveraging large-scale pre-training (e.g., (112)) and data augmentation to more fundamental approaches like invariant representation learning (e.g., (25)) and explicit adaptation to dynamic shifts (e.g., (228)). A key tension lies between learning highly specific, powerful representations for a given task and ensuring these representations transfer effectively to novel settings. The problem of homophily/heterophily also exemplifies a structural distribution shift that GNNs must contend with. Future research needs to focus on developing more principled causal inference techniques for graphs, designing pre-training objectives that lead to truly transferable knowledge, and creating adaptive GNN architectures that can learn and evolve in dynamic, open-world environments. The interplay with robust evaluation (Section 6.4) is crucial, as reliable generalization can only be confirmed through rigorous and realistic benchmarking.

The Need for Robust Evaluation and Benchmarking

The rapid proliferation of Graph Neural Network architectures and applications has inadvertently created a significant challenge: the lack of robust, standardized evaluation methodologies and benchmarking frameworks (44; 216). This issue, highlighted as a critical pitfall in works like (5), hinders scientific progress, makes fair comparisons between models difficult, and compromises the reproducibility and trustworthiness of research findings. Without consistent and realistic evaluation, it becomes challenging to discern genuinely superior approaches from those that merely benefit from favorable experimental setups or outdated baselines.

Context and Motivation: The GNN literature is replete with new models claiming state-of-the-art performance, but these claims are often based on inconsistent data splits, varying hyperparameter tuning efforts, or comparisons against suboptimally implemented baselines. For instance, (5) critically examines link prediction, revealing that many existing baselines are underreported due to poor hyperparameter tuning or non-standard settings. This creates a misleading landscape where true progress is obscured. Furthermore, evaluation settings often fail to reflect real-world scenarios, such as using "easy" negative samples in link prediction that are trivial to classify, as pointed out by (5). The

absence of standardized benchmarks for emerging challenges like dynamic graphs, heterogeneous graphs (140; 211), or specific application domains (e.g., brain networks (27; 280)) further exacerbates this problem.

Method Family A: Standardized Benchmarking Frameworks and Platforms

* **Problem Solved:** These initiatives aim to provide a common ground for evaluating GNNs, ensuring consistency, reproducibility, and fair comparisons across different models and research groups. They address the practical limitation of fragmented and inconsistent evaluation practices.

* **Core Innovation & Mechanism:**

* **Open Graph Benchmark (OGB):** OGB (27) is a prominent example, providing a collection of diverse, large-scale, and realistic graph datasets with standardized data splits, evaluation metrics, and leaderboards. This allows researchers to compare their models against a common set of challenges.

* **Domain-Specific Benchmarks:** Beyond general benchmarks, specialized platforms like BrainGB (27) for brain network analysis, PowerGraph (58) for power grid datasets, and FedGraphNN (185) for federated learning on graphs, cater to unique challenges and data characteristics of specific domains.

* **Benchmarking Graph Neural Networks (44):** This work provides a comprehensive benchmarking effort, evaluating a wide range of GNN architectures on diverse tasks and datasets, aiming to establish robust baselines and identify effective design principles.

* **Conditions for Success:** Benchmarks succeed when they are widely adopted by the community, offer diverse and challenging tasks, and maintain up-to-date baselines.

* **Theoretical Limitations:** Even well-designed benchmarks are finite and cannot cover all possible graph structures or real-world complexities. They might implicitly favor certain types of GNN architectures or inductive biases.

* **Practical Limitations:** Creating and maintaining large-scale, high-quality benchmarks is resource-intensive. Benchmarks can quickly become outdated as new GNN architectures and capabilities emerge. The sheer diversity of graph types and tasks makes a single "universal" benchmark impractical.

* **Comparison:** OGB (27) provides broad coverage across various tasks and scales. Domain-specific benchmarks (e.g., BrainGB (27)) offer deeper insights into particular application areas. The work by (44) provides a systematic evaluation of existing GNNs, complementing the

dataset-focused OGB. This evolution shows a growing recognition of the need for both general and specialized evaluation platforms.

Method Family B: Critical Analysis of Evaluation Pitfalls and New Benchmarking Strategies * **Problem Solved:** This family critically examines existing evaluation practices, identifies their shortcomings, and proposes more rigorous and realistic evaluation strategies to ensure fair and meaningful comparisons. * **Core Innovation & Mechanism:** * **Identifying Pitfalls in Link Prediction Evaluation:** (5) provides a seminal critique of GNN evaluation for link prediction. Their work systematically re-evaluates 17 prominent methods across 7 datasets, revealing that many models achieve significantly higher performance than previously reported when properly tuned and evaluated under consistent settings. They highlight issues such as inconsistent data splits, non-standard metrics, and unrealistic negative sampling. ***Heuristic Related Sampling Technique (HeaRT):** To address the unrealistic negative sampling in link prediction, (5) proposes HeaRT, a novel technique that generates hard*, heuristic-related negative samples. This makes the evaluation task more challenging and reflective of real-world scenarios where negative links are often structurally similar to positive ones.* * **Evaluation of Trustworthiness Aspects:** Beyond predictive accuracy, there's a growing need to evaluate other dimensions of trustworthy GNNs (as discussed in Section 5). This includes evaluating explainability (43; 76), robustness to attacks (36; 82; 317; 306; 262), and fairness (66). * **Noisy Label Benchmarks:** (318) introduces NoisyGL, a comprehensive benchmark specifically designed for evaluating GNNs under label noise, addressing a common imperfection in real-world graph datasets. * **Conditions for Success:** Critical analyses succeed when their findings are reproducible and lead to widespread adoption of improved practices. New evaluation strategies like HeaRT (5) require community consensus and integration into standard benchmarks. * **Theoretical Limitations:** Defining "realism" in evaluation can be subjective and domain-dependent. Developing universal metrics for complex attributes like explainability or fairness remains an open theoretical challenge. * **Practical Limitations:** Implementing rigorous hyperparameter searches for all baselines can be computationally expensive. Adopting new, more challenging eval-

uation settings might initially show lower performance for existing models, which can be a barrier to adoption. **Comparison:** *While Method Family A provides the platforms, Method Family B provides the critical scrutiny and methodological innovation for evaluation. (5) exemplifies how a deep dive into specific evaluation pitfalls can profoundly reshape understanding of model performance. This evolution signifies a maturation of the field, moving beyond simply reporting numbers to critically questioning how* those numbers are generated.*

Synthesis and Implications: The need for robust evaluation and benchmarking is paramount for the continued progress and credibility of GNN research. The field is actively addressing this through the development of standardized platforms (e.g., OGB (?)) and critical analyses that expose and rectify evaluation pitfalls (e.g., (5)). A persistent tension exists between the desire for simple, fast evaluation and the need for realistic, comprehensive assessment. Most papers implicitly assume that higher numbers on a benchmark directly translate to real-world utility, but this assumption is increasingly being questioned. Future research must focus on developing more diverse and complex benchmarks that reflect real-world challenges (e.g., dynamic, heterogeneous, multi-modal graphs), creating standardized codebases to enhance reproducibility, and establishing community-driven platforms for continuous evaluation and comparison. Furthermore, integrating the evaluation of trustworthiness dimensions (explainability, fairness, privacy) into standard benchmarking is crucial for fostering responsible GNN development.

7. Applications and Real-World Impact

Graph Neural Networks (GNNs) have emerged as a transformative paradigm in machine learning, extending the power of deep learning to non-Euclidean, graph-structured data. Their inherent ability to model complex relationships and dependencies within interconnected datasets has led to a profound and diverse real-world impact across an ever-expanding array of domains (146; 42; 235). This section highlights the versatility of GNNs in handling intricate data structures, showcasing their successful application in critical areas ranging from commercial systems to scientific discovery and societal chal-

lenges. The widespread adoption of GNNs in these fields is a testament to their improved expressive power (as discussed in Section 3), enhanced robustness (Section 4), and the continuous efforts to address fundamental challenges such as scalability, generalization, and trustworthiness (Section 6 and Section 5).

The utility of GNNs stems from their capacity to learn rich, context-aware representations of nodes and edges by iteratively aggregating information from local neighborhoods. This message-passing mechanism, while facing challenges like over-smoothing and over-squashing (Section 6.2), allows GNNs to capture both local structural patterns and global graph properties, which are crucial for understanding complex systems. For instance, in recommender systems, GNNs move beyond simple pairwise interactions to model high-order user-item relationships and social influence, leading to more accurate and personalized suggestions. In scientific domains, they provide unprecedented tools for predicting molecular properties, discovering new materials, and analyzing intricate brain connectivity patterns, accelerating research and development. Furthermore, GNNs' capacity for predictive learning in dynamic environments is evident in urban computing, time series analysis, and epidemic modeling, where they process spatio-temporal data to forecast future states and inform decision-making.

However, the application of GNNs in these real-world settings is not without its complexities, often exposing new facets of the challenges discussed in previous sections. For example, the scalability concerns raised in Section 6.1 are particularly acute in web-scale recommender systems with billions of users and items, necessitating distributed training and efficient sampling strategies. Similarly, the demand for high expressiveness (Section 3) and interpretability (Section 5) becomes paramount in scientific and medical applications, where understanding the underlying mechanisms is as important as predictive accuracy. The challenge of generalization to unseen structures and distribution shifts (Section 6.3) is critical in dynamic environments like urban traffic or epidemic spread, where models must adapt to constantly evolving patterns. Finally, the growing role of GNNs in sensitive areas like cybersecurity and financial fraud detection underscores the urgent need for robust (Section 4) and trustworthy (Section 5) models that can withstand adversarial

attacks and provide fair, explainable outcomes. This section delves into these diverse applications, illustrating how GNNs are being tailored and advanced to meet the unique demands of each domain, while simultaneously pushing the boundaries of graph-based machine learning.

GNNs in Recommender Systems

Recommender systems are ubiquitous in modern digital platforms, guiding users through vast selections of products, services, and content. Traditional recommendation approaches, such as collaborative filtering and matrix factorization, often struggle with data sparsity, the cold-start problem (recommending for new users or items), and capturing the complex, high-order interactions inherent in user behavior (4; 89). Graph Neural Networks offer a natural and powerful framework for addressing these limitations, by explicitly modeling the intricate relationships within user-item interaction graphs and leveraging auxiliary information like social connections or knowledge graphs (92; 155). The field has seen a rapid evolution, moving from basic GNN applications to highly specialized architectures designed for dynamic, personalized, and context-aware recommendations.

Method Family A: Collaborative Filtering and User-Item Interaction Modeling This family of approaches focuses on enhancing recommendation accuracy by directly modeling the user-item interaction graph.

- **Problem Solved:** The core problem addressed is the sparse and complex nature of user-item interactions, which traditional methods struggle to fully leverage for accurate predictions. GNNs aim to capture high-order connectivity patterns that imply latent preferences.
- **Core Innovation & Mechanism:** Early pioneering work, such as (111), introduced Graph Convolutional Neural Networks (GCNs) to model user-item bipartite graphs. The innovation lies in propagating information across this graph, allowing user embeddings to be enriched by the items they interact with, and item embeddings by the users who consume them. This iterative message passing effectively

captures collaborative signals. Subsequent works, like LightGCN (??), demonstrated that simpler GNN architectures, by removing non-linearities and feature transformations and focusing purely on neighborhood aggregation, could achieve superior performance. This highlights a key insight: for recommendation, the graph structure itself often carries more signal than complex feature transformations.

- **Evidence:** (111) reported substantial improvements in recall and precision on large-scale datasets, such as Google Play, demonstrating the practical efficacy of GNNs in a web-scale setting. Models like LightGCN have consistently achieved state-of-the-art results on various e-commerce and content recommendation benchmarks.
- **Limitations:** Despite their success, these models face significant practical limitations, particularly concerning scalability for massive user-item graphs with billions of nodes and edges, a challenge explicitly discussed in Section 6.1. The computational cost of full-graph convolution can be prohibitive, necessitating sampling strategies or distributed training frameworks (198; 171). Furthermore, while GNNs mitigate the cold-start problem to some extent by leveraging structural information, they do not fully resolve it for entirely new users or items without any historical interactions.
- **Comparison:** GNNs offer a more principled and expressive way to model high-order interactions compared to traditional matrix factorization methods, which often rely on latent factors and struggle to capture complex, non-linear relationships. They provide richer, context-aware representations than simple embedding methods by explicitly incorporating the graph topology. However, their computational overhead is generally higher than simpler factorization models.

Method Family B: Session-Based and Sequential Recommendation This family focuses on capturing dynamic user preferences within a single interaction session or over a sequence of interactions.

- **Problem Solved:** The challenge here is to model the evolving, short-term interests of a user within a session, as well as the sequential dependencies between items. Tra-

ditional methods often treat sessions as independent or struggle with the dynamic nature of preferences.

- **Core Innovation & Mechanism:** (127) pioneered the application of GNNs to session-based recommendation, modeling each session as a graph where items are nodes and sequential interactions form directed edges. The GNN then learns session-level representations by aggregating item embeddings. Building on this, (35) provides a comprehensive survey of GNNs in sequential recommendation, highlighting various architectural choices. More recent works, such as (182) and (223), have explored dynamic GNNs and attention mechanisms to capture fine-grained temporal dynamics and personalized preferences within sessions, enhancing the model’s ability to predict the next item. (136) further introduced global context enhancement to improve session-based recommendations.
- **Evidence:** These approaches have consistently demonstrated improved next-item prediction accuracy on e-commerce and streaming service datasets, outperforming RNN-based and other sequential models.
- **Limitations:** Handling long-term dependencies and mitigating catastrophic forgetting (a challenge for dynamic GNNs, as noted in Section 6.3 and (197)) in evolving user preferences remains a significant hurdle. Real-time updates for dynamic sessions are computationally intensive, posing practical challenges for high-throughput systems. The information loss inherent in some GNN designs for session-based recommendation also needs careful consideration (153).
- **Comparison:** GNNs offer a more structured and flexible way to model session dynamics than traditional recurrent neural networks (RNNs/LSTMs), which might struggle with complex, non-linear item transitions and the non-sequential nature of some session graphs. GNNs can capture both direct item-item transitions and the broader global context of a session more effectively.

Method Family C: Social Recommendation and Knowledge-Aware GNNs

This family leverages auxiliary information, such as social connections and external knowledge graphs, to enrich recommendations.

- **Problem Solved:** These methods address data sparsity and enhance recommendation quality by incorporating external signals beyond direct user-item interactions. Social recommendation aims to leverage peer influence, while knowledge-aware GNNs enrich item representations with semantic information.
- **Core Innovation & Mechanism:** (79) demonstrated the utility of GNNs for social recommendation, modeling user-user social graphs to capture influence and homophily among connected users. This allows for recommendations based on trusted connections. (155) provides a survey on this area. Knowledge-aware GNNs, exemplified by (137), integrate external knowledge graphs (e.g., item attributes, categories, relationships) into the recommendation process. By treating the knowledge graph as a heterogeneous graph, GNNs can propagate information from entities to items, providing richer semantic context and mitigating sparsity. More recently, (212) focused on knowledge-enhanced GNNs for explainable recommendation, aiming to provide not just predictions but also reasons behind them, addressing a key trustworthiness concern (Section 5).
- **Evidence:** These approaches have shown enhanced recommendation accuracy, particularly in scenarios with sparse interaction data, and improved interpretability for users.
- **Limitations:** The use of social network data raises significant privacy concerns (Section 5), requiring careful anonymization and ethical considerations. Integrating heterogeneous information from diverse knowledge graphs can be complex and computationally challenging, as highlighted by surveys on GNNs for knowledge graphs (206).
- **Comparison:** Social GNNs explicitly model peer influence, a factor often missed by traditional collaborative filtering. Knowledge-aware GNNs provide richer semantic context for items, which is a significant advantage over purely interaction-based

models, especially for cold-start items. This integration of diverse graph structures (user-item, user-user, item-entity) exemplifies the versatility of GNNs in handling multi-relational data.

In synthesis, GNNs have profoundly transformed recommender systems by offering a flexible and powerful framework for modeling complex, high-order relationships. The field is actively evolving towards more dynamic, personalized, and explainable systems, often leveraging large-scale pre-training strategies (as discussed in Section 6.3) and distributed architectures (Section 6.1) to handle the immense scale of web-based data (171). However, the inherent tension between achieving high scalability, maintaining expressive power, and ensuring trustworthy (fair, private, explainable) recommendations remains a central and active area of research.

Scientific Domains: Molecules, Materials, and Brain Networks

The scientific enterprise is fundamentally driven by the discovery and understanding of complex systems, often characterized by intricate interactions and non-Euclidean structures. Graph Neural Networks are uniquely positioned to accelerate this discovery process by providing powerful tools for modeling these inherent relationships in domains such as molecular science, materials discovery, and brain network analysis (51; 167). By representing entities (e.g., atoms, brain regions) as nodes and their interactions (e.g., chemical bonds, functional connectivity) as edges, GNNs can learn structure-function relationships that are difficult for traditional methods to capture, leading to new insights and predictive capabilities.

Method Family A: Molecular Science and Drug Discovery This family applies GNNs to understand and predict properties of molecules, crucial for drug development.

- **Problem Solved:** The primary problem is to accurately predict various molecular properties (e.g., toxicity, solubility, binding affinity, ADMET properties) and to design novel molecules with desired characteristics. Traditional methods often rely on hand-crafted molecular descriptors, which can be incomplete or lack generalizability.

- Core Innovation & Mechanism:** Molecules are inherently graph-structured, with atoms as nodes and chemical bonds as edges. GNNs learn molecular representations (or "fingerprints") directly from this graph structure, capturing both atomic features and their topological arrangement. (165) provided an early comparison, demonstrating the superiority of graph-based models over descriptor-based ones for drug discovery tasks. (188) introduced structure-aware interactive GNNs for predicting protein-ligand binding affinity, modeling the complex interface between two molecular graphs. A significant innovation comes from equivariant GNNs, such as GemNet (99) and the E(3)-equivariant GNNs by (124). These models incorporate geometric information and respect physical symmetries (e.g., rotation, translation invariance), which is crucial for accurately modeling 3D molecular structures and interatomic potentials. The recent Mole-BERT (169) explores pre-training GNNs for molecules, aiming to learn generalizable representations from large unlabeled datasets, addressing the generalization challenge (Section 6.3). Furthermore, (271) demonstrates the use of attention-based GNNs for ADMET (Absorption, Distribution, Metabolism, Excretion, Toxicity) property prediction, highlighting the role of attention in focusing on key molecular substructures.
- Evidence:** GNNs have achieved state-of-the-art accuracy in predicting a wide range of molecular properties, accelerating the virtual screening and design phases of drug discovery. For instance, E(3)-equivariant GNNs have shown high accuracy in predicting interatomic forces, crucial for molecular dynamics simulations (124).
- Limitations:** The expressiveness of GNNs (Section 3) for distinguishing complex substructures, such as those related to chirality, remains a challenge. Interpretability (Section 5) is paramount for scientific validation and regulatory approval, but GNNs can often act as black boxes (8; 28). Data scarcity for novel compounds is also a practical limitation.
- Comparison:** GNNs offer a direct, end-to-end learning approach that bypasses the need for extensive, often incomplete, hand-crafted molecular descriptors, a key ad-

vantage over traditional cheminformatics methods. Equivariant GNNs (e.g., (124)) are particularly powerful as they inherently respect the underlying physical symmetries, providing a strong inductive bias that leads to more accurate and data-efficient models compared to non-equivariant GNNs.

Method Family B: Materials Discovery and Engineering This family leverages GNNs to predict and design materials with desired properties.

- **Problem Solved:** The challenge is to predict various material properties (e.g., stability, mechanical strength, electronic properties, defect formation energy) and to accelerate the discovery of new materials with tailored characteristics. Traditional experimental and computational (e.g., DFT) methods are often time-consuming and expensive.
- **Core Innovation & Mechanism:** Similar to molecules, materials (e.g., crystals, amorphous solids) can be represented as graphs, where atoms are nodes and inter-atomic bonds are edges. GNNs learn the complex structure-property relationships. (51) provides a comprehensive survey on the application of GNNs in materials science and chemistry. (175) benchmarks various GNN architectures for materials chemistry tasks, highlighting their relative strengths. (205) demonstrated GNNs’ ability to predict stress, strain, and deformation fields in materials, crucial for mechanical engineering. More recently, (298) leveraged persistent homology features within GNNs for accurate defect formation energy predictions, showcasing the integration of topological data analysis with GNNs.
- **Evidence:** GNNs have shown promising results in accelerating materials design cycles, reducing the need for extensive experimental validation, and improving the accuracy of property predictions.
- **Limitations:** Data availability remains a significant practical limitation, as experimental data for novel materials is expensive to obtain. Generalizability to entirely new material classes (a facet of the generalization challenge discussed in Section 6.3)

is also a concern. Incorporating multi-scale information, from atomic interactions to macroscopic material behavior, remains an open problem.

- **Comparison:** GNNs provide a data-driven approach to materials science, complementing or even surpassing traditional computational methods like Density Functional Theory (DFT) for certain predictions, especially when large datasets are available. They offer a more flexible framework for exploring the vast chemical space than rule-based or descriptor-based methods.

Method Family C: Brain Network Analysis This family applies GNNs to understand the human brain’s connectivity and function.

- **Problem Solved:** The core problem is to understand complex brain connectivity patterns (connectomes), diagnose neurological and psychiatric disorders (e.g., Alzheimer’s disease, epilepsy), predict disease progression, and evaluate treatment responses. Traditional statistical methods often struggle with the high-dimensionality and complex topology of brain networks.
- **Core Innovation & Mechanism:** Brain activity data (e.g., fMRI, EEG) can be represented as graphs, where brain regions are nodes and functional or structural connections are edges. GNNs are then used to analyze these "brain graphs" to identify disease-specific patterns. (93) provides a survey of GNNs in network neuroscience. BrainGB (27) serves as a critical benchmark for brain network analysis with GNNs, addressing the need for robust evaluation (Section 6.4). A key innovation is the focus on interpretable GNNs for connectome-based brain disorder analysis, as demonstrated by (39), which is crucial for clinical adoption and scientific discovery (Section 5). (236) further explores deep reinforcement learning guided GNNs for brain network analysis, showcasing advanced learning paradigms. More recently, (278) applied GNNs for Alzheimer’s disease and epilepsy classification, while (305) focused on comorbidity-based frameworks for Alzheimer’s. (280) provides a comprehensive overview of methods, challenges, and future directions in this rapidly evolving field.

- **Evidence:** GNNs have shown improved accuracy in classifying brain disorders, identifying potential biomarkers, and providing insights into the neurobiological underpinnings of disease.
- **Limitations:** Brain network datasets are often small and heterogeneous, posing challenges for robust model training and generalization (Section 6.3). Interpretability is paramount for clinical adoption, requiring GNNs that can provide clear, biologically meaningful explanations. The dynamic nature of brain activity (temporal graphs) is complex to model, a challenge highlighted in surveys on temporal GNNs (33).
- **Comparison:** GNNs offer a powerful way to leverage the topological and functional information embedded in brain networks, which traditional machine learning methods often struggle to capture. The explicit focus on interpretability (e.g., (39)) is a critical differentiator for clinical applications, where black-box models are unacceptable.

In synthesis, GNNs are becoming indispensable tools in scientific discovery, enabling data-driven insights into complex systems across molecular, materials, and neuroscientific domains. The demand for high expressiveness (Section 3) to capture intricate structural details and for interpretability (Section 5) to foster scientific understanding is particularly acute in these fields. The development of specialized GNNs that respect physical symmetries (e.g., equivariant GNNs) or leverage advanced mathematical concepts like fractional calculus (FROND (3), which mitigates over-smoothing by modeling memory-dependent dynamics) represents a key trend. However, the challenges of small, high-dimensional datasets, the need for robust generalization (Section 6.3), and the integration of domain-specific knowledge remain significant areas for future research.

Urban Computing, Time Series, and Epidemic Modeling

Many critical real-world phenomena, from traffic patterns to disease outbreaks, exhibit complex spatio-temporal dependencies. These systems are characterized by dynamic in-

teractions between interconnected entities over time, making them ideal candidates for modeling with Graph Neural Networks. GNNs, particularly their spatio-temporal variants, have proven highly effective in capturing these intricate dynamics, enabling predictive learning in complex and constantly evolving environments (30; 237; 110). This section explores their transformative impact in urban computing, multivariate time series analysis, and epidemic modeling.

Method Family A: Urban Computing and Intelligent Transportation Systems This family focuses on optimizing urban infrastructure and services through predictive modeling.

- **Problem Solved:** The core problem is to accurately predict dynamic urban phenomena such as traffic flow, travel times, public transport demand, and ride-sharing availability. These predictions are vital for urban planning, smart navigation, and efficient resource allocation.
- **Core Innovation & Mechanism:** Urban environments are naturally represented as graphs (e.g., roads as edges, intersections or regions as nodes). Spatio-temporal GNNs (STGNNs) are designed to capture both spatial correlations (e.g., how traffic congestion propagates through a network) and temporal dynamics (e.g., daily and weekly patterns, sudden changes). (130) introduced Spatial-Temporal Fusion Graph Neural Networks for traffic flow forecasting, demonstrating how to integrate both dimensions effectively. A prominent real-world application is Google Maps’ ETA prediction, where GNNs play a crucial role in estimating travel times by modeling road networks and traffic conditions (16). (170) provides a comprehensive survey of GNNs for intelligent transportation systems. More recently, (263) proposed a generic integration paradigm of topology-free patterns for traffic speed prediction, highlighting the ongoing innovation in handling complex urban dynamics.
- **Evidence:** STGNNs have consistently shown superior accuracy in traffic prediction and ETA estimation compared to traditional statistical and machine learning methods, leading to more efficient navigation and better urban planning decisions.

- **Limitations:** Real-time processing of massive, continuously streaming sensor data from city-wide networks poses significant scalability challenges (Section 6.1). Handling unforeseen events (e.g., accidents, extreme weather) that disrupt learned patterns and lead to distribution shifts (Section 6.3) remains difficult.
- **Comparison:** STGNNs significantly outperform traditional time series models (e.g., ARIMA, LSTMs) by explicitly modeling spatial dependencies, which are critical for interconnected urban systems. They are also more flexible than grid-based Convolutional Neural Networks (CNNs) for handling the irregular topologies of road networks.

Method Family B: Time Series Analysis (Forecasting, Imputation, Anomaly Detection) This family extends GNNs to general multivariate time series tasks.

- **Problem Solved:** This family addresses the challenges of forecasting future values in multivariate time series, imputing missing data, and detecting anomalies in interconnected sensor networks or financial markets. Traditional methods often treat individual time series independently or struggle to capture complex inter-series dependencies.
- **Core Innovation & Mechanism:** In these applications, individual time series (e.g., from different sensors, stock prices) are represented as nodes in a graph, with edges representing learned correlations or physical connections. GNNs then learn the inter-series dependencies and propagate information across the graph to enhance predictions. (126) demonstrated how GNNs can connect the dots for multivariate time series forecasting. (161) successfully applied GNNs for multivariate time series imputation, effectively filling data gaps by leveraging contextual information from related series. (142) rethought GNNs for anomaly detection, identifying unusual patterns in interconnected time series. (30) provides a comprehensive survey of GNNs for various time series tasks. Recent work by (249) introduces causality-aware spatio-temporal GNNs for imputation, emphasizing the importance of causal relationships for robust predictions. (243) applies deep learning-based STGNNs for

price movement classification in crude oil and precious metal markets, showcasing their utility in financial forecasting.

- **Evidence:** GNN-based methods have shown enhanced accuracy in forecasting, robust imputation of missing values, and effective detection of anomalies in diverse datasets, from environmental sensors to financial indicators.
- **Limitations:** A key challenge is defining the optimal graph structure (e.g., how to infer meaningful edges from raw time series data when no explicit graph exists). Handling highly non-stationary time series and providing explainability for detected anomalies remain open problems.
- **Comparison:** GNNs provide a powerful inductive bias for multivariate time series by explicitly modeling the relationships between series, which traditional methods often treat independently or only implicitly through complex feature engineering. They offer a more holistic view of the system’s dynamics.

Method Family C: Epidemic Modeling This family focuses on predicting and understanding disease spread.

- **Problem Solved:** The critical problem is to accurately predict disease spread, identify potential outbreak hotspots, and evaluate the effectiveness of various intervention strategies. Traditional epidemiological models (e.g., SIR, SEIR) often simplify population interactions, limiting their granularity.
- **Core Innovation & Mechanism:** Human contact networks, geographical regions, or transportation networks can be naturally modeled as graphs. GNNs learn the complex dynamics of disease transmission over these networks, capturing how infections propagate through interconnected populations. (196) proposed CausalGNN, a causal-based GNN for spatio-temporal epidemic forecasting, emphasizing the importance of causal relationships for understanding and predicting disease spread. (69) provides a comprehensive review of GNNs in epidemic modeling, highlighting the diverse approaches and challenges.

- **Evidence:** GNNs have demonstrated improved accuracy in forecasting infection rates, identifying vulnerable populations, and simulating the impact of interventions, providing valuable insights for public health policy.
- **Limitations:** Data privacy concerns (Section 5) are paramount when dealing with individual-level contact data. The dynamic and uncertain nature of human behavior, as well as the rapid evolution of pathogens, pose significant challenges for model generalizability across different diseases and regions (Section 6.3).
- **Comparison:** GNNs offer a more realistic and granular way to model disease spread than traditional compartmental models (e.g., SIR, SEIR) by explicitly incorporating network structure and individual-level interactions. They can capture complex, non-linear transmission dynamics that are difficult to represent otherwise.

In summary, GNNs have proven highly effective in modeling complex spatio-temporal phenomena, moving beyond static graph analysis to capture dynamic processes. The field is actively integrating GNNs with causal inference (e.g., (196; 249)) to improve predictive power and provide actionable insights, addressing the need for explainability and trustworthiness (Section 5). However, the challenges of scalability to large, continuously evolving graphs (Section 6.1) and robust generalization to unseen events or novel disease strains (Section 6.3) are particularly critical and remain active areas of research in these domains.

GNNs in Cybersecurity and Other Emerging Fields

Beyond the well-established applications, Graph Neural Networks are increasingly being adopted in a diverse array of emerging fields, demonstrating their broad utility in analyzing complex, interconnected data for specialized tasks. These applications span critical areas like cybersecurity, where relational patterns are key to detecting threats, to resource management and various other domains that benefit from graph-structured data analysis. This expansion underscores GNNs' fundamental versatility and their capacity to provide powerful solutions where traditional methods fall short.

Method Family A: Cybersecurity and Fraud Detection This family leverages GNNs to identify and mitigate various forms of digital threats and financial malfeasance.

- **Problem Solved:** The core problem is to detect sophisticated cyber threats (e.g., network intrusions, malware, phishing, software vulnerabilities) and financial fraud (e.g., credit card fraud, money laundering) by analyzing complex relational data. Traditional signature-based or rule-based systems are often reactive and struggle with novel attacks.
- **Core Innovation & Mechanism:** Cybersecurity data naturally forms graphs: network traffic as communication graphs, system calls as execution graphs, financial transactions as monetary flow graphs, and code dependencies as program graphs. GNNs are used to identify anomalous patterns, malicious entities, or vulnerable structures within these graphs. (21) provides an overview of GNNs aiding defensive cyber operations. (191) surveys GNNs for intrusion detection, highlighting their ability to learn attack patterns. (132)’s Design pioneered the use of GNNs for vulnerability identification by learning comprehensive program semantics. (147) combined GNNs with expert knowledge for smart contract vulnerability detection, demonstrating the power of hybrid approaches. (180) applied GNNs for accurate decentralized application identification via encrypted traffic analysis. In financial fraud, (246) enhanced credit card fraud detection using causal temporal GNNs, emphasizing the importance of causal relationships in identifying fraudulent activities. (203) even explores quantum GNNs for financial fraud detection, pointing towards future directions. (321) discusses the role of GNNs in threat intelligence knowledge graphs, showcasing their utility in organizing and reasoning over vast amounts of threat data.
- **Evidence:** GNNs have shown improved detection rates for various cyber threats and fraud types, often outperforming traditional machine learning models by capturing subtle relational anomalies.
- **Limitations:** Adversarial robustness (Section 4) is paramount in cybersecurity, as

attackers can actively manipulate graph structures to evade detection (36; 82; 317; 306; 262). Interpretability (Section 5) is crucial for forensic analysis and understanding attack vectors. Handling concept drift in attack patterns and the sheer volume of real-time data are also significant practical challenges.

- **Comparison:** GNNs offer a more holistic view of cyber threats by modeling relationships (e.g., attack propagation paths, dependency chains) that traditional rule-based or feature-engineering methods often miss. They can detect novel attacks by learning anomalous graph structures, providing a significant advantage over signature-based systems.

Method Family B: Resource Management and Optimization This family applies GNNs to optimize resource allocation and system performance in complex networks.

- **Problem Solved:** The problem is to optimize resource allocation in dynamic, large-scale systems such as wireless communication networks, power grids, and healthcare facilities. These tasks often involve complex combinatorial optimization problems that are difficult to solve efficiently.
- **Core Innovation & Mechanism:** Communication networks (e.g., base stations, users), power grids (e.g., generators, transmission lines), or healthcare systems (e.g., patients, doctors, resources) can all be modeled as graphs. GNNs learn optimal allocation strategies or predict system states to improve efficiency and stability. (149) and (154) discuss the application of GNNs for scalable radio resource management, optimizing spectrum allocation and interference control. (172) reviews GNNs in power systems, highlighting their use for stability analysis and fault detection. (204) learned power allocation for multi-cell-multi-user systems using heterogeneous GNNs. (258) applied GNNs for resource allocation optimization in the healthcare industry. More recently, (276) introduced physics-informed GNNs for water distribution systems, integrating physical laws into the learning process. (291) explored power control for 6G in-factory subnetworks using GNNs.
- **Evidence:** GNNs have demonstrated more efficient resource utilization, improved

system stability, reduced operational costs, and faster decision-making in these complex domains.

- **Limitations:** Real-time decision-making under dynamic conditions and strict latency constraints is challenging. Integrating complex physics-informed constraints (e.g., power flow equations in electrical grids) effectively remains an active research area. Scalability to extremely large infrastructure networks (Section 6.1) is also a concern.
- **Comparison:** GNNs provide a data-driven approach to optimization problems that are often intractable with traditional combinatorial optimization methods, especially in dynamic, large-scale settings. They can learn complex, non-linear relationships between system components and adapt to changing conditions.

Method Family C: Other Emerging Applications This broad category encompasses a wide range of novel applications that leverage GNNs’ ability to process graph-structured data.

- **Problem Solved:** This includes diverse problems such as object detection and multi-object tracking in computer vision (176; 125), text classification and knowledge graph completion in natural language processing (41; 162; 206), anomaly detection in Industrial IoT (78; 186), and even more specialized tasks like fault diagnosis in machinery (163; 304; 309), brain network analysis for medical diagnosis (39; 278), and quantum computing (203; 324).
- **Core Innovation & Mechanism:** GNNs are adapted to these diverse data types by constructing appropriate graph representations. For instance, in computer vision, GNNs can model relationships between objects in a scene (7) or points in a point cloud (266). In NLP, they model word dependencies or knowledge graph structures (177; 73). The key innovation is the flexible application of the message-passing paradigm to extract meaningful features from non-Euclidean data across various domains.

- **Evidence:** These applications demonstrate the expanding applicability of deep learning to previously challenging non-Euclidean data, often achieving state-of-the-art performance in their respective fields.
- **Limitations:** Each domain presents unique challenges, including data representation issues (how to best construct the graph), the need for specialized GNN architectures, and domain-specific evaluation metrics. Generalizability (Section 6.3) across different sub-tasks within a domain can also be an issue.
- **Comparison:** GNNs offer a powerful paradigm shift for many fields by enabling the direct modeling of relationships, which was previously difficult or required extensive, often suboptimal, feature engineering. This allows for a more holistic and context-aware understanding of the data.

In conclusion, the widespread adoption of GNNs across diverse and emerging fields underscores their fundamental versatility and growing importance as a foundational technology for analyzing interconnected data. Cybersecurity applications highlight the critical need for robust (Section 4) and interpretable (Section 5) GNNs, especially when operating in adversarial environments. Resource management applications demonstrate GNNs' capacity for complex optimization and predictive control. The continuous emergence of new applications, from quantum computing to industrial fault diagnostics, indicates that GNNs are pushing the boundaries of what is possible with deep learning. The ongoing challenges of generalization (Section 6.3) and scalability (Section 6.1) are consistently encountered and addressed through domain-specific innovations, driving further research and development in this dynamic field.

8. Conclusion: Future Directions and Ethical Considerations

The journey through Graph Neural Networks (GNNs) has revealed a rapidly evolving field, transitioning from foundational message-passing paradigms to highly specialized ar-

chitectures capable of addressing complex, real-world challenges. As evidenced in Section 7, GNNs have demonstrated transformative potential across diverse applications, from enhancing recommender systems and accelerating scientific discovery to bolstering cybersecurity and optimizing urban infrastructure. This success, however, is not without its underlying complexities and unresolved tensions, many of which were explored in previous sections concerning expressiveness (Section 3), robustness (Section 4), trustworthiness (Section 5), and fundamental challenges like scalability and generalization (Section 6). This concluding section synthesizes the current state of GNN research, casting a forward-looking perspective on its future trajectory. It identifies emerging trends and novel paradigms that promise to push the boundaries of graph learning, such as multi-modal integration with large language models and the adoption of advanced mathematical frameworks like fractional calculus. Concurrently, it emphasizes the ongoing imperative to bridge theoretical advancements with practical deployment, addressing persistent challenges like scalability, generalizability, and the inherent trade-offs between model power and efficiency. Finally, it reiterates the critical ethical considerations and the urgent need for developing responsible AI, ensuring that GNNs are not only powerful and efficient but also fair, transparent, and privacy-preserving, thereby guiding future research towards impactful and ethical innovation for societal benefit.

The evolution of GNNs has been characterized by a continuous quest for greater expressive power and efficiency. Early GNNs, while groundbreaking, were often limited by the Weisfeiler-Lehman (WL) test, struggling to distinguish certain non-isomorphic graphs (122; 65). This led to the development of more sophisticated architectures that incorporate higher-order information or structural features, as discussed in Section 3. Simultaneously, the challenge of deploying GNNs on massive, real-world graphs necessitated innovations in scalability, moving from full-graph processing to sampling-based and distributed training strategies (Section 6.1). The increasing depth of GNNs, while desirable for complex tasks, introduced issues like over-smoothing and over-squashing, where node representations become indistinguishable or information bottlenecks emerge (133; 84; 59). These persistent challenges underscore a recurring tension: enhancing expressive power often comes at the

cost of increased computational complexity, while simplifying models for scalability can compromise their ability to capture intricate graph structures. The future of GNNs lies in navigating these trade-offs through novel architectural designs, advanced mathematical tools, and a holistic approach that integrates performance with ethical considerations. The emergence of multi-modal GNNs, which combine graph data with other rich information sources like text or images, represents a significant paradigm shift, promising to unlock new levels of understanding and application. However, as GNNs become more pervasive and powerful, particularly in sensitive domains like healthcare, finance, and cybersecurity, the ethical implications—fairness, transparency, and privacy—transition from secondary concerns to foundational requirements for responsible AI development, demanding rigorous attention from the research community.

Emerging Trends and Novel Paradigms

The future trajectory of Graph Neural Networks is marked by a profound shift towards more sophisticated architectures and integrated learning paradigms, moving beyond the traditional message-passing framework to address inherent limitations and unlock new capabilities. This evolution is driven by the need for enhanced expressiveness, better handling of complex data, and the ability to model intricate real-world dynamics. Two major emerging trends are the multi-modal integration with large language models (LLMs) and the adoption of advanced mathematical frameworks, alongside architectural innovations like non-convolutional and path-based GNNs.

Method Family A: Multi-modal Integration with Large Language Models

This family explores the synergistic combination of GNNs with LLMs to leverage both structural and semantic information.

- **Problem Solved:** GNNs excel at capturing structural relationships but often lack deep semantic understanding, while LLMs possess vast world knowledge and reasoning capabilities but struggle with explicit graph structures. The problem is to bridge this gap, allowing GNNs to benefit from rich textual semantics and LLMs to gain structural context. This is particularly relevant for tasks like knowledge

graph completion, text classification, and recommendation where both structural and semantic cues are vital (41; 162).

- **Core Innovation & Mechanism:** The innovation lies in developing architectures that enable seamless information exchange between GNNs and LLMs. One approach involves using LLMs to enrich node features with semantic embeddings, providing a more informative input to GNNs. Conversely, GNNs can provide structural context to LLMs, guiding their reasoning over knowledge graphs or document networks. (77) introduced GraphPrompt, unifying pre-training and downstream tasks for GNNs by leveraging prompt tuning, a technique popularized by LLMs, to adapt pre-trained GNNs to new tasks with minimal data. Similarly, (166) proposed multi-task prompting for GNNs, enabling a single model to handle diverse graph tasks by conditioning on task-specific prompts. (75) explored universal prompt tuning for GNNs, demonstrating its efficacy in various settings. A more direct integration is seen in (73), which investigates if GNNs can learn language with extremely weak text supervision, suggesting a deep interplay. Recent works, such as (287), propose Hybrid-LLM-GNNs for enhanced materials property prediction, where LLMs provide high-level textual descriptions and GNNs process molecular structures. (250) even explores whether LLMs can improve the adversarial robustness of GNNs, indicating a broader role for LLMs in GNN trustworthiness. (239) incorporates syntax and semantics with dual GNNs for aspect-level sentiment analysis, showcasing how GNNs can be designed to process linguistic structures.
- **Evidence:** Preliminary results indicate that combining GNNs with LLMs can significantly improve performance on tasks requiring both structural and semantic understanding, such as knowledge graph reasoning, text classification, and multi-modal recommendation. For instance, prompt-tuned GNNs have shown competitive performance with fewer labeled examples, mirroring LLM efficiency.
- **Limitations:** A critical challenge is the alignment of heterogeneous representations from graphs and text, ensuring that the fused information is coherent and

meaningful. The computational cost of integrating large LLMs with GNNs can be substantial, especially for large graphs. Furthermore, the interpretability of such complex multi-modal models becomes even more challenging.

- **Comparison:** This approach moves beyond simple concatenation of features (e.g., using BERT embeddings as node features) towards deeper, interactive learning paradigms. Unlike traditional GNNs that primarily rely on structural aggregation, or LLMs that operate on sequential tokens, multi-modal GNN-LLM models aim for a holistic understanding that leverages the strengths of both, offering a more comprehensive representation of complex entities and their relationships.

Method Family B: Advanced Mathematical Frameworks and Architectural Innovations This family introduces novel mathematical foundations and architectural designs to overcome fundamental GNN limitations.

- **Problem Solved:** This addresses long-standing GNN limitations such as limited expressive power (e.g., inability to distinguish WL-indistinguishable graphs), over-smoothing (node representations becoming identical with depth), over-squashing (difficulty in propagating information over long distances), and the lack of memory in continuous GNNs.
- **Core Innovation & Mechanism:**
 - * **Fractional Calculus:** (3) introduced FROND, a framework that incorporates the Caputo fractional derivative into continuous GNNs. This is a core innovation because it allows GNNs to model non-Markovian, memory-dependent dynamics, where the current state depends on the entire history of node features, not just the immediate past. This inherently mitigates over-smoothing by leading to an algebraic (slower) rate of convergence to stationarity, unlike the exponential convergence of integer-order models.
 - * **Non-Convolutional GNNs:** (1) proposed RUM (Random Walk with Unifying Memory), an entirely convolution-free GNN. Its innovation lies in processing graph information via stochastic random walks and RNNs, using a novel "anonymous experiment" to encode topological environments. This de-

sign theoretically and empirically addresses limited expressiveness (surpassing 1-WL), over-smoothing (non-diminishing Dirichlet energy), and over-squashing (slower inter-node Jacobian decay) simultaneously, offering a fundamentally different approach to graph learning. * **Path-Based GNNs:** (6) introduced Path Neural Networks (PathNNs) that explicitly leverage path information. Their key innovation is operating on "annotated sets of paths," recursively enriching path information to achieve expressive power beyond 1-WL, and even distinguishing graphs that are 3-WL indistinguishable. This directly tackles the expressiveness bottleneck by focusing on richer structural primitives. * **Geometric Equivariant GNNs:** These models (e.g., (102; 124; 226; 14)) are designed to respect physical symmetries (e.g., rotation, translation) inherent in 3D data like molecules or point clouds. Their innovation is to build in these inductive biases, leading to more data-efficient and accurate models for geometric tasks. (300) improved equivariant GNNs on large geometric graphs via virtual nodes learning, while (285) questioned assumptions about high-degree representations in these models.

- **Evidence:** FROND (3) demonstrated consistent performance improvements over integer-order continuous GNNs across various datasets. RUM (1) achieved competitive performance while being faster and more scalable than many convolutional GNNs, with empirical validation of its theoretical benefits in expressiveness and over-smoothing mitigation. PathNNs (6) empirically showed their ability to distinguish graphs beyond 3-WL, validating their enhanced expressive power. Equivariant GNNs have achieved state-of-the-art results in molecular property prediction and interatomic potential modeling (124).
- **Limitations:** Fractional calculus introduces theoretical complexity and relies on numerical solvers, which can have computational implications. Non-convolutional and path-based GNNs, while expressive, can face challenges with computational cost for very long paths or extremely dense graphs, although RUM explicitly addresses this. Geometric GNNs are specialized for 3D data

and may not directly apply to all graph types.

- **Comparison:** This family represents a fundamental rethinking of GNNs. Unlike methods that incrementally improve message passing, these approaches either introduce entirely new mathematical operators (fractional calculus), abandon the convolution paradigm altogether (non-convolutional GNNs), or leverage richer structural primitives (paths) to achieve higher expressive power. They offer deeper, more principled solutions to core GNN limitations, moving beyond architectural tweaks to foundational changes. The theoretical work on the expressive power of pooling (90) and path-based GNNs (301) further underscores this shift.

Bridging Theory and Practice

The rapid empirical success of Graph Neural Networks has often outpaced a comprehensive theoretical understanding, leading to a significant gap between what models achieve and why they achieve it. Bridging this theory-practice divide is crucial for developing more robust, generalizable, and efficient GNNs that can reliably operate in real-world, large-scale, and dynamic environments. This involves addressing persistent challenges like scalability and generalization, while simultaneously deepening the theoretical foundations that underpin GNN design.

Method Family A: Scalability for Large-Scale Graphs This family focuses on developing GNNs that can efficiently process and learn from massive graphs, often with billions of nodes and edges.

- * **Problem Solved:** As discussed in Section 6.1, traditional GNNs suffer from high memory consumption and computational cost when applied to large graphs, primarily due to the need to access and process entire neighborhoods or even the full adjacency matrix. This limits their deployment in web-scale applications like recommender systems (171) or social networks.
- * **Core Innovation & Mechanism:** Innovations center on reducing the

computational burden while preserving performance. * **Sampling Strategies:** Techniques like neighbor sampling (e.g., GraphSAGE (16)) and subgraph sampling reduce the size of the computational graph per node. * **Distributed Training:** Frameworks like DistGNN (198) enable parallel processing of large graphs across multiple machines, distributing the computational load. * **Efficient Architectures:** Models like SIGN (63) and GNNAutoScale (201) leverage pre-computation of fixed-depth neighborhood aggregations or historical embeddings to reduce real-time computation. (168) scaled GNNs with approximate PageRank, demonstrating efficiency. * **Graph Condensation:** (119) proposed graph condensation to create smaller, synthetic graphs that retain the essential information of large original graphs, significantly reducing training time. * **Non-convolutional Designs:** As highlighted by (1), non-convolutional GNNs like RUM can be inherently more scalable, with runtime complexity agnostic to the number of edges, making them suitable for huge graphs without requiring all neighbors to be present. * **Lightweight Frameworks:** (325) introduced TGLite, a lightweight programming framework for continuous-time temporal GNNs, addressing scalability for dynamic graphs.

* **Evidence:** Distributed GNNs have shown significant speedups (e.g., 3x faster training for large graphs (198)) and reduced memory footprints, enabling training on graphs with billions of edges. Graph condensation can reduce training time by orders of magnitude while maintaining competitive accuracy. RUM (1) empirically demonstrated faster runtime than even simple convolutional GNNs.

* **Limitations:** Sampling strategies can introduce bias and may not always capture the full context of a node’s neighborhood. Distributed training adds system complexity. Graph condensation can lead to information loss, and its effectiveness depends on the condensation algorithm. The trade-off

between accuracy and efficiency remains a persistent challenge.

- * **Comparison:** The evolution of scalability solutions has moved from heuristic sampling to more principled distributed training and, more recently, to architectural innovations (e.g., non-convolutional designs) and data reduction techniques (graph condensation) that inherently improve efficiency. This reflects a shift from external optimization to internal architectural design for scalability.

Method Family B: Generalization and Robustness This family addresses the critical need for GNNs to perform reliably on unseen data and to withstand adversarial manipulations.

- **Problem Solved:** GNNs often struggle to generalize to graphs with different distributions or unseen structures (Section 6.3), and they are vulnerable to adversarial attacks that subtly perturb graph structure or features to induce misclassification (Section 4). This poses a significant barrier to their deployment in critical applications like cybersecurity (36).
- **Core Innovation & Mechanism:**
 - * **Pre-training and Prompt Tuning:** Inspired by NLP, pre-training GNNs on large unlabeled graphs using self-supervised objectives (e.g., (49; 20)) improves generalization. Subsequent prompt tuning (15; 77) adapts these pre-trained models to downstream tasks efficiently.
 - * **Data Augmentation and Graph Rewiring:** Techniques like DropEdge (?) (stochastically removing edges) or more sophisticated graph rewiring strategies (97; 257) enhance robustness by exposing the model to diverse graph structures during training. (13) surveyed data augmentation for GNNs.
 - * **Invariant Learning and Causal GNNs:** Methods that learn invariant representations (88) or incorporate causal inference (e.g., causal GNNs (25)) aim to make models robust to spurious correlations and distribution shifts.
 - * **Adversarial Training and Certified Robust-**

ness: Adversarial training (82; 104; 87) explicitly trains GNNs against attacks. More recently, certified robustness methods (262) provide formal guarantees on a GNN’s resistance to perturbations within a defined bound. (40) focused on robust GNNs for noisy graphs with sparse labels. * **Homophily/Heterophily Adaptation:** The ongoing debate about homophily (Section 6.2, (72; 128; 217; 113; 108)) has led to adaptive GNNs that can handle both homophilous and heterophilous graphs, improving generalization across diverse graph types (2; 208). GloGNN (2) is a prime example, finding global homophily in heterophilous settings by learning signed coefficient matrices and achieving linear time complexity for global aggregation, significantly improving performance and efficiency.

- **Evidence:** Pre-trained GNNs have shown significant performance gains on downstream tasks, especially with limited labeled data. Adversarial training and certified defenses have demonstrated improved resilience against various attack types, with (262) offering deterministic certification. GloGNN (2) achieved superior performance and efficiency on heterophilous graphs, demonstrating the grouping effect of its global aggregation.
- **Limitations:** Pre-training requires large datasets and computational resources. Certified robustness is often limited to specific attack models and perturbation budgets. Defining and achieving "invariance" for complex graph data is challenging. The trade-off between robustness and accuracy often exists.
- **Comparison:** The field is moving from reactive, empirical defenses against specific attacks to proactive, theoretically grounded approaches for generalization and certified robustness. This involves a shift from simply improving accuracy to ensuring reliable performance under diverse and challenging conditions. The work by (2) exemplifies this

by providing a principled approach to a long-standing challenge (heterophily) with theoretical backing and practical efficiency.

Method Family C: Deeper Theoretical Foundations This family focuses on developing a more rigorous mathematical understanding of GNNs’ capabilities and limitations.

- **Problem Solved:** Despite empirical success, the theoretical underpinnings of GNNs, particularly regarding their expressive power, generalization capabilities, and failure modes (e.g., over-smoothing, over-squashing), are still being fully elucidated. This lack of theory hinders principled model design and reliable deployment.
- **Core Innovation & Mechanism:** Innovations involve applying advanced mathematical tools to analyze GNNs.
 - * **Expressive Power Analysis:** Beyond the 1-WL test (122), research explores higher-order WL tests (65), spectral analysis (53; 189), and the expressive power of specific architectural components like pooling (90) or paths (6; 301). (160) offered a new perspective on how GNNs go beyond WL. (251) explored counting graph substructures with GNNs, a task related to expressive power (60).
 - * **Generalization Bounds:** Developing PAC-Bayesian bounds and other statistical learning theory frameworks to understand and predict GNN generalization performance (55; 68; 265).
 - * **Understanding Limitations:** Deeper analysis of over-smoothing (81; 115; 59; 230; 282) and over-squashing (84) provides insights into their root causes and informs mitigation strategies. (1) provides theoretical analysis of how non-convolutional GNNs alleviate these issues.
 - * **Unifying Frameworks:** Efforts to unify GNNs with optimization frameworks (85) or to connect them to differential equations (199) and even logic (299) aim to provide a more coherent theoretical understanding. (118) provides a comprehensive theory of GNNs.
- **Evidence:** Theoretical proofs have established the expressive power of

certain GNN architectures, guiding the design of more powerful models. Analysis of over-smoothing has led to architectural modifications like residual connections and normalization. Generalization bounds provide a theoretical basis for understanding why GNNs perform well on unseen data.

- **Limitations:** Theoretical analysis often requires simplifying assumptions that may not hold in complex real-world graphs. Bridging the gap between theoretical guarantees and empirical performance remains challenging. The mathematical complexity can be high, limiting accessibility.

- **Comparison:** This area represents a shift from purely empirical model development to a more principled, theory-driven approach. It aims to provide the "why" behind GNN behavior, informing better architectural choices and hyperparameter tuning, and ultimately leading to more reliable and predictable models. The work by (323) specifically aims to bridge generalization and expressivity.

Ethical Considerations and Responsible AI Development

As Graph Neural Networks become increasingly integrated into critical societal infrastructures and decision-making processes, the ethical implications of their deployment demand paramount attention. The imperative for developing responsible AI is no longer a peripheral concern but a foundational requirement, ensuring that GNNs are not only powerful but also fair, transparent, and privacy-preserving. This involves addressing inherent biases, providing meaningful explanations, and safeguarding sensitive information against malicious actors.

Method Family A: Fairness and Bias Mitigation This family focuses on identifying and mitigating biases in GNNs to ensure equitable

outcomes.

· **Problem Solved:** GNNs, like other AI models, can learn and amplify biases present in the training data, leading to unfair or discriminatory predictions, especially when sensitive attributes (e.g., gender, race) are correlated with graph structure or features. This is a critical concern in applications like loan default prediction (272) or social recommendation. As highlighted in Section 5, fairness is a key aspect of trustworthiness.

· **Core Innovation & Mechanism:** Innovations involve developing methods to detect, measure, and mitigate bias. * **Debiasing Techniques:** (11) proposed EDITS to model and mitigate data bias for GNNs. (98) focused on learning fair GNNs with limited sensitive attribute information, acknowledging the practical constraints of data availability. * **Fairness-Aware GNNs:** (9) introduced a ranking-based approach for individual fairness in GNNs, ensuring similar individuals receive similar outcomes. (116) aimed to improve fairness by mitigating sensitive attribute leakage, preventing models from implicitly using protected characteristics. * **Disentangled Causal Substructure Learning:** (25) proposed debiasing GNNs via learning disentangled causal substructures, separating causal factors from confounding biases. (12) rethought fair GNNs from a re-balancing perspective, adjusting data distributions to reduce bias. * **Structural Disparity Analysis:** (103) demystified structural disparity in GNNs, questioning if one size fits all for fairness, highlighting the need for context-aware solutions. (322) proposed FUGNN, harmonizing fairness and utility in GNNs, addressing the common trade-off.

· **Evidence:** These methods have demonstrated success in reducing disparate impact and improving various fairness metrics (e.g., demographic parity, equal opportunity) while maintaining competitive util-

ity on benchmark datasets. For example, debiasing techniques can reduce performance gaps between different demographic groups by a significant margin.

- **Limitations:** Defining and measuring fairness in complex graph data is inherently challenging and context-dependent. There are often trade-offs between fairness and model utility (accuracy), requiring careful balancing. Detecting subtle biases, especially those encoded implicitly in graph topology, remains difficult.

- **Comparison:** Early work focused on identifying bias; recent efforts are moving towards proactive mitigation strategies, including architectural changes (disentangled learning) and data-centric approaches (re-balancing), often seeking to balance fairness with other performance objectives.

Method Family B: Transparency and Explainability This family aims to make GNN predictions understandable and interpretable to humans.

- **Problem Solved:** GNNs are often black-box models, making it difficult to understand *why* a particular prediction was made. This lack of transparency hinders trust, accountability, and debugging, especially in high-stakes applications like medical diagnosis (39) or vulnerability detection (173). As discussed in Section 5, explainability is crucial for trustworthiness.

- **Core Innovation & Mechanism:** Innovations focus on generating human-intelligible explanations. * **Post-hoc Explainers:** Methods like GNNExplainer (31), XGNN (19), and PGM-Explainer (91) identify important nodes, edges, or features that contribute to a prediction. (105) provides a taxonomic survey of explainability in GNNs. **Sub-graph Explanations:** (8) introduced *SubgraphX*, a novel method that directly identifies important subgraphs* as explanations, which are of-

ten more intuitive than individual nodes/edges. This is a significant advancement as it focuses on connected, meaningful patterns. (28) proposed explaining GNNs via structure-aware interaction index, while (38) focused on inductive and efficient explanations. (67) aims to unveil global interactive patterns for interpretable GNNs. (264) introduced GOAt for Graph Output Attribution.

- * **Counterfactual Explanations:** (107) developed CF-GNNExplainer, which identifies minimal changes to the input graph that flip a prediction, providing insights into decision boundaries.
- * **Inherently Interpretable GNNs:** Some approaches aim to build GNNs that are interpretable by design, such as ProtGNN (54) or KerGNNs (215) that use graph kernels. (64) focused on discovering invariant rationales.
- * **Evaluation of Explainability:** (43) and (76) critically evaluate how interpretable GNN explanations truly are, highlighting the ongoing challenges in this area. (212) focused on knowledge-enhanced GNNs for explainable recommendation.
- **Evidence:** Explainers have successfully identified critical graph components for various tasks, helping users understand model decisions. SubgraphX (8) demonstrated significantly improved explanations compared to node/edge-level methods, providing more intuitive and human-intelligible insights.
- **Limitations:** The fidelity of explanations (how accurately they reflect the model's true reasoning) can be limited. Explanations can be computationally expensive to generate, especially for large graphs. The subjective nature of "interpretability" means that what constitutes a good explanation can vary across users and domains.
- **Comparison:** The field has progressed from simple feature attribution to more complex, structural (subgraph, path) and counterfactual explanations. There's a growing emphasis on human-centered evaluation of explanations, moving beyond purely quantitative metrics to

assess their utility for domain experts.

Method Family C: Privacy and Security This family addresses the vulnerabilities of GNNs to privacy breaches and adversarial attacks.

- **Problem Solved:** GNNs can inadvertently leak sensitive information about individuals or organizations present in the graph (e.g., node features, connectivity patterns). Furthermore, they are susceptible to various adversarial attacks (e.g., poisoning, evasion, backdoor attacks) that can compromise their integrity and reliability, as discussed in Section 4 and Section 5.
- **Core Innovation & Mechanism:** Innovations focus on developing privacy-preserving training methods and robust defense mechanisms. * **Privacy-Preserving GNNs:** Federated learning for GNNs (185; 195) allows models to be trained on decentralized data without sharing raw graph information, addressing privacy concerns. Differential privacy techniques can be applied to GNN training to protect sensitive node features or graph structures. * **Adversarial Defenses:** Research focuses on making GNNs robust against various attacks. (74) demonstrated link stealing attacks, while (26) showed backdoor attacks. Defenses include adversarial training (104; 82), certified robustness (262), and architectural modifications like GNNGuard (83). (36) critically assessed the robustness of GNN defenses. (109) explored unnoticeable backdoor attacks, highlighting the evolving threat landscape. (317) and (306) explored topological adversarial attacks and bounding expected robustness. * **Trustworthy GNN Frameworks:** Surveys like (61) and (66) provide comprehensive overviews of trustworthiness aspects, including privacy, robustness, fairness, and explainability, guiding integrated solutions. (94) focused on confidence calibration for trustworthy GNNs.

- **Evidence:** Federated GNNs have shown the ability to train models collaboratively while preserving data locality and privacy. Robustness techniques have significantly reduced the success rate of various adversarial attacks, with certified defenses providing quantifiable guarantees.
- **Limitations:** Implementing differential privacy often comes with a trade-off in model utility. Federated learning introduces communication overhead and challenges in handling heterogeneous data distributions. The arms race between attackers and defenders is continuous, requiring constant innovation in defense mechanisms.
- **Comparison:** The field has matured from identifying vulnerabilities (e.g., link stealing, backdoor attacks) to developing sophisticated, multi-faceted defense strategies, including privacy-by-design principles (federated learning) and theoretically grounded robustness guarantees (certified defenses). This reflects a holistic approach to securing GNNs across their lifecycle.

In conclusion, the future of Graph Neural Networks is bright, characterized by a dynamic interplay between theoretical advancements, practical innovations, and a growing commitment to ethical development. The emerging trends, from multi-modal integration with LLMs to the adoption of fractional calculus and novel non-convolutional architectures, promise to unlock unprecedented capabilities, addressing long-standing limitations in expressiveness, scalability, and the modeling of complex dynamics. Simultaneously, the persistent challenges of bridging theory and practice—particularly in achieving robust generalization and efficient scalability for real-world deployment—continue to drive fundamental research. Crucially, as GNNs become more powerful and pervasive, the ethical considerations of fairness, transparency, and privacy are no longer optional but integral to responsible AI development. The ongoing research in debiasing, explainability, and robust

security measures reflects a collective commitment to ensuring that GNNs serve as powerful tools for societal benefit, guiding future innovation towards impactful and ethical solutions.

References

References

- [1] Yuanqing Wang, and Kyunghyun Cho (2024). *Non-convolutional Graph Neural Networks*. Neural Information Processing Systems.
- [2] Xiang Li, Renyu Zhu, Yao Cheng, et al. (2022). *Finding Global Homophily in Graph Neural Networks When Meeting Heterophily*. International Conference on Machine Learning.
- [3] Qiyu Kang, Kai Zhao, Qinxu Ding, et al. (2024). *Unleashing the Potential of Fractional Calculus in Graph Neural Networks with FROND*. International Conference on Learning Representations.
- [4] Chen Gao, Xiang Wang, Xiangnan He, et al. (2022). *Graph Neural Networks for Recommender System*. Web Search and Data Mining.
- [5] Juanhui Li, Harry Shomer, Haitao Mao, et al. (2023). *Evaluating Graph Neural Networks for Link Prediction: Current Pitfalls and New Benchmarking*. Neural Information Processing Systems.
- [6] Gaspard Michel, Giannis Nikolentzos, J. Lutzeyer, et al. (2023). *Path Neural Networks: Expressive and Accurate Graph Neural Networks*. International Conference on Machine Learning.
- [7] Chaoqi Chen, Yushuang Wu, Qiyuan Dai, et al. (2022). *A Survey on Graph Neural Networks and Graph Transformers in Computer Vision: A Task-Oriented Perspective*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [8] Hao Yuan, Haiyang Yu, Jie Wang, et al. (2021). *On Explainabil-*

- ity of Graph Neural Networks via Subgraph Explorations*. International Conference on Machine Learning.
- [9] Yushun Dong, Jian Kang, H. Tong, et al. (2021). *Individual Fairness for Graph Neural Networks: A Ranking based Approach*. Knowledge Discovery and Data Mining.
- [10] Quentin Cappart, D. Chételat, Elias Boutros Khalil, et al. (2021). *Combinatorial optimization and reasoning with graph neural networks*. International Joint Conference on Artificial Intelligence.
- [11] Yushun Dong, Ninghao Liu, B. Jalaeian, et al. (2021). *EDITS: Modeling and Mitigating Data Bias for Graph Neural Networks*. The Web Conference.
- [12] Zhixun Li, Yushun Dong, Qiang Liu, et al. (2024). *Rethinking Fair Graph Neural Networks from Re-balancing*. Knowledge Discovery and Data Mining.
- [13] Tong Zhao, Yozen Liu, Leonardo Neves, et al. (2020). *Data Augmentation for Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [14] Chaitanya K. Joshi, and Simon V. Mathis (2023). *On the Expressive Power of Geometric Graph Neural Networks*. International Conference on Machine Learning.
- [15] Mingchen Sun, Kaixiong Zhou, Xingbo He, et al. (2022). *GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [16] Austin Derrow-Pinion, Jennifer She, David Wong, et al. (2021). *ETA Prediction with Graph Neural Networks in Google Maps*.

International Conference on Information and Knowledge Management.

- [17] Yu Chen, Lingfei Wu, and Mohammed J. Zaki (2020). *Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings*. Neural Information Processing Systems.
- [18] Hanqing Zeng, Muhan Zhang, Yinglong Xia, et al. (2022). *Decoupling the Depth and Scope of Graph Neural Networks*. Neural Information Processing Systems.
- [19] Haonan Yuan, Jiliang Tang, Xia Hu, et al. (2020). *XGNN: Towards Model-Level Explanations of Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [20] Yaochen Xie, Zhao Xu, Zhengyang Wang, et al. (2021). *Self-Supervised Learning of Graph Neural Networks: A Unified Review*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [21] Shaswata Mitra, Trisha Chakraborty, Subash Neupane, et al. (2024). *Use of Graph Neural Networks in Aiding Defensive Cyber Operations*. arXiv.org.
- [22] Muhan Zhang, and Pan Li (2021). *Nested Graph Neural Networks*. Neural Information Processing Systems.
- [23] Hongya Wang, Haoteng Yin, Muhan Zhang, et al. (2022). *Equivariant and Stable Positional Encoding for More Powerful Graph Neural Networks*. International Conference on Learning Representations.

- [24] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, et al. (2021). *Learning to Pre-train Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [25] Shaohua Fan, Xiao Wang, Yanhu Mo, et al. (2022). *Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure*. Neural Information Processing Systems.
- [26] Zaixi Zhang, Jinyuan Jia, Binghui Wang, et al. (2020). *Backdoor Attacks to Graph Neural Networks*. ACM Symposium on Access Control Models and Technologies.
- [27] Hejie Cui, Wei Dai, Yanqiao Zhu, et al. (2022). *BrainGB: A Benchmark for Brain Network Analysis With Graph Neural Networks*. IEEE Transactions on Medical Imaging.
- [28] Ngoc Bui, Hieu Trung Nguyen, Viet Anh Nguyen, et al. (2024). *Explaining Graph Neural Networks via Structure-aware Interaction Index*. International Conference on Machine Learning.
- [29] Chuang Liu, Yibing Zhan, Chang Li, et al. (2022). *Graph Pooling for Graph Neural Networks: Progress, Challenges, and Opportunities*. International Joint Conference on Artificial Intelligence.
- [30] Ming Jin, Huan Yee Koh, Qingsong Wen, et al. (2023). *A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [31] Rex Ying, Dylan Bourgeois, Jiaxuan You, et al. (2019). *GNNExplainer: Generating Explanations for Graph Neural Networks*. Neural Information Processing Systems.
- [32] Meng Liu, Hongyang Gao, and Shuiwang Ji (2020). *Towards*

Deeper Graph Neural Networks. Knowledge Discovery and Data Mining.

- [33] Antonio Longa, Veronica Lachi, G. Santin, et al. (2023). *Graph Neural Networks for temporal graphs: State of the art, open challenges, and opportunities*. Trans. Mach. Learn. Res..
- [34] P. Papp, Karolis Martinkus, Lukas Faber, et al. (2021). *DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks*. Neural Information Processing Systems.
- [35] Jianxin Chang, Chen Gao, Y. Zheng, et al. (2021). *Sequential Recommendation with Graph Neural Networks*. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
- [36] Felix Mujkanovic, Simon Geisler, Stephan Gunnemann, et al. (2023). *Are Defenses for Graph Neural Networks Robust?*. Neural Information Processing Systems.
- [37] Jiaxuan You, Jonathan M. Gomes-Selman, Rex Ying, et al. (2021). *Identity-aware Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [38] Dongsheng Luo, Tianxiang Zhao, Wei Cheng, et al. (2024). *Towards Inductive and Efficient Explanations for Graph Neural Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [39] Hejie Cui, Wei Dai, Yanqiao Zhu, et al. (2022). *Interpretable Graph Neural Networks for Connectome-Based Brain Disorder Analysis*. International Conference on Medical Image Computing and Computer-Assisted Intervention.

- [40] Enyan Dai, Wei-dong Jin, Hui Liu, et al. (2022). *Towards Robust Graph Neural Networks for Noisy Graphs with Sparse Labels*. Web Search and Data Mining.
- [41] Kunze Wang, Yihao Ding, and S. Han (2023). *Graph Neural Networks for Text Classification: A Survey*. Artificial Intelligence Review.
- [42] Bharti Khemani, S. Patil, K. Kotecha, et al. (2024). *A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions*. Journal of Big Data.
- [43] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, et al. (2022). *Evaluating explainability for graph neural networks*. Scientific Data.
- [44] Vijay Prakash Dwivedi, Chaitanya K. Joshi, T. Laurent, et al. (2023). *Benchmarking Graph Neural Networks*. Journal of machine learning research.
- [45] Ralph Abboud, .Ismail .Ilkan Ceylan, Martin Grohe, et al. (2020). *The Surprising Power of Graph Neural Networks with Random Node Initialization*. International Joint Conference on Artificial Intelligence.
- [46] Yixin Liu, Kaize Ding, Jianling Wang, et al. (2023). *Learning Strong Graph Neural Networks with Weak Information*. Knowledge Discovery and Data Mining.
- [47] Xiaorui Liu, W. Jin, Yao Ma, et al. (2021). *Elastic Graph Neural Networks*. International Conference on Machine Learning.
- [48] M. Balcilar, P. Héroux, Benoit Gaüzère, et al. (2021). *Breaking*

the Limits of Message Passing Graph Neural Networks. International Conference on Machine Learning.

- [49] Weihua Hu, Bowen Liu, Joseph Gomes, et al. (2019). *Strategies for Pre-training Graph Neural Networks*. International Conference on Learning Representations.
- [50] B. Chamberlain, S. Shirobokov, Emanuele Rossi, et al. (2022). *Graph Neural Networks for Link Prediction with Subgraph Sketching*. International Conference on Learning Representations.
- [51] Patrick Reiser, Marlen Neubert, Andr’e Eberhard, et al. (2022). *Graph neural networks for materials science and chemistry*. Communications Materials.
- [52] Guohao Li, Matthias Müller, Bernard Ghanem, et al. (2021). *Training Graph Neural Networks with 1000 Layers*. International Conference on Machine Learning.
- [53] Xiyuan Wang, and Muhan Zhang (2022). *How Powerful are Spectral Graph Neural Networks*. International Conference on Machine Learning.
- [54] Zaixin Zhang, Qi Liu, Hao Wang, et al. (2021). *ProtGNN: Towards Self-Explaining Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [55] Vikas K. Garg, S. Jegelka, and T. Jaakkola (2020). *Generalization and Representational Limits of Graph Neural Networks*. International Conference on Machine Learning.
- [56] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi (2021). *SLAPS: Self-Supervision Improves Structure Learning for Graph Neural Networks*. Neural Information Processing Systems.

- [57] Yuyu Zhang, Xinshi Chen, Yuan Yang, et al. (2021). *Graph Neural Networks*. Deep Learning on Graphs.
- [58] Anna Varbella, Kenza Amara, B. Gjorgiev, et al. (2024). *PowerGraph: A power grid benchmark dataset for graph neural networks*. Neural Information Processing Systems.
- [59] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra (2023). *A Survey on Oversmoothing in Graph Neural Networks*. arXiv.org.
- [60] Zhengdao Chen, Lei Chen, Soledad Villar, et al. (2020). *Can graph neural networks count substructures?*. Neural Information Processing Systems.
- [61] He Zhang, Bang Wu, Xingliang Yuan, et al. (2022). *Trustworthy Graph Neural Networks: Aspects, Methods, and Trends*. Proceedings of the IEEE.
- [62] Haoyu Han, Juanhui Li, Wei Huang, et al. (2024). *Node-wise Filtering in Graph Neural Networks: A Mixture of Experts Approach*. arXiv.org.
- [63] Emanuele Rossi, Fabrizio Frasca, B. Chamberlain, et al. (2020). *SIGN: Scalable Inception Graph Neural Networks*. arXiv.org.
- [64] Yingmin Wu, Xiang Wang, An Zhang, et al. (2022). *Discovering Invariant Rationales for Graph Neural Networks*. International Conference on Learning Representations.
- [65] Christopher Morris, Martin Ritzert, Matthias Fey, et al. (2018). *Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks*. AAAI Conference on Artificial Intelligence.

- [66] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, et al. (2022). *A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability*. Machine Intelligence Research.
- [67] Yuwen Wang, Shunyu Liu, Tongya Zheng, et al. (2024). *Unveiling Global Interactive Patterns across Graphs: Towards Interpretable Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [68] Haotian Ju, Dongyue Li, Aneesh Sharma, et al. (2023). *Generalization in Graph Neural Networks: Improved PAC-Bayesian Bounds on Graph Diffusion*. International Conference on Artificial Intelligence and Statistics.
- [69] Zewen Liu, Guancheng Wan, B. A. Prakash, et al. (2024). *A Review of Graph Neural Networks in Epidemic Modeling*. Knowledge Discovery and Data Mining.
- [70] Muhan Zhang, and Yixin Chen (2018). *Link Prediction Based on Graph Neural Networks*. Neural Information Processing Systems.
- [71] F. Bianchi, Daniele Grattarola, L. Livi, et al. (2019). *Graph Neural Networks With Convolutional ARMA Filters*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [72] Yao Ma, Xiaorui Liu, Neil Shah, et al. (2021). *Is Homophily a Necessity for Graph Neural Networks?*. International Conference on Learning Representations.
- [73] Li, Lecheng Zheng, Bowen Jin, et al. (2024). *Can Graph Neural Networks Learn Language with Extremely Weak Text Supervision?*. Annual Meeting of the Association for Computational Linguistics.

- [74] Xinlei He, Jinyuan Jia, M. Backes, et al. (2020). *Stealing Links from Graph Neural Networks*. USENIX Security Symposium.
- [75] Taoran Fang, Yunchao Zhang, Yang Yang, et al. (2022). *Universal Prompt Tuning for Graph Neural Networks*. Neural Information Processing Systems.
- [76] Yongqiang Chen, Yatao Bian, Bo Han, et al. (2024). *How Interpretable Are Interpretable Graph Neural Networks?*. International Conference on Machine Learning.
- [77] Zemin Liu, Xingtong Yu, Yuan Fang, et al. (2023). *GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks*. The Web Conference.
- [78] Guimin Dong, Mingyue Tang, Zhiyuan Wang, et al. (2022). *Graph Neural Networks in IoT: A Survey*. ACM Trans. Sens. Networks.
- [79] Wenqi Fan, Yao Ma, Qing Li, et al. (2019). *Graph Neural Networks for Social Recommendation*. The Web Conference.
- [80] Jiaxuan You, Rex Ying, and J. Leskovec (2020). *Design Space for Graph Neural Networks*. Neural Information Processing Systems.
- [81] Chen Cai, and Yusu Wang (2020). *A Note on Over-Smoothing for Graph Neural Networks*. arXiv.org.
- [82] Lukas Gosch, Simon Geisler, Daniel Sturm, et al. (2023). *Adversarial Training for Graph Neural Networks*. Neural Information Processing Systems.
- [83] Xiang Zhang, and M. Zitnik (2020). *GNNGuard: Defending Graph Neural Networks against Adversarial Attacks*. Neural Information Processing Systems.

- [84] Uri Alon, and Eran Yahav (2020). *On the Bottleneck of Graph Neural Networks and its Practical Implications*. International Conference on Learning Representations.
- [85] Meiqi Zhu, Xiao Wang, C. Shi, et al. (2021). *Interpreting and Unifying Graph Neural Networks with An Optimization Framework*. The Web Conference.
- [86] Xu Zou, Qinkai Zheng, Yuxiao Dong, et al. (2021). *TDGIA: Effective Injection Attacks on Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [87] Kaidi Xu, Hongge Chen, Sijia Liu, et al. (2019). *Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective*. International Joint Conference on Artificial Intelligence.
- [88] Donglin Xia, Xiao Wang, Nian Liu, et al. (2024). *Learning Invariant Representations of Graph Neural Networks via Cluster Generalization*. Neural Information Processing Systems.
- [89] Shiwen Wu, Fei Sun, Fei Sun, et al. (2020). *Graph Neural Networks in Recommender Systems: A Survey*. ACM Computing Surveys.
- [90] F. Bianchi, and Veronica Lachi (2023). *The expressive power of pooling in Graph Neural Networks*. Neural Information Processing Systems.
- [91] Minh N. Vu, and M. Thai (2020). *PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks*. Neural Information Processing Systems.
- [92] Chen Gao, Yu Zheng, Nian Li, et al. (2021). *A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions*. Trans. Recomm. Syst..

- [93] Alaa Bessadok, M. Mahjoub, and I. Rekik (2021). *Graph Neural Networks in Network Neuroscience*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [94] Xiao Wang, Hongrui Liu, Chuan Shi, et al. (2021). *Be Confident! Towards Trustworthy Graph Neural Networks via Confidence Calibration*. Neural Information Processing Systems.
- [95] Simon Geisler, Arthur Kosmala, Daniel Herbst, et al. (2024). *Spatio-Spectral Graph Neural Networks*. Neural Information Processing Systems.
- [96] DingYi Zeng, Wanlong Liu, Wenyu Chen, et al. (2023). *Substructure Aware Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [97] Wei Jin, Yao Ma, Xiaorui Liu, et al. (2020). *Graph Structure Learning for Robust Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [98] Enyan Dai, and Suhang Wang (2020). *Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information*. Web Search and Data Mining.
- [99] Johannes Klicpera, Florian Becker, and Stephan Gunnemann (2021). *GemNet: Universal Directional Graph Neural Networks for Molecules*. Neural Information Processing Systems.
- [100] Vijay Prakash Dwivedi, A. Luu, T. Laurent, et al. (2021). *Graph Neural Networks with Learnable Structural and Positional Representations*. International Conference on Learning Representations.

- [101] Jiarui Feng, Yixin Chen, Fuhai Li, et al. (2022). *How Powerful are K -hop Message Passing Graph Neural Networks*. Neural Information Processing Systems.
- [102] Victor Garcia Satorras, Emiel Hoogeboom, and M. Welling (2021). *$E(n)$ Equivariant Graph Neural Networks*. International Conference on Machine Learning.
- [103] Haitao Mao, Zhikai Chen, Wei Jin, et al. (2023). *Demystifying Structural Disparity in Graph Neural Networks: Can One Size Fit All?*. Neural Information Processing Systems.
- [104] Daniel Zügner, and Stephan Günnemann (2019). *Adversarial Attacks on Graph Neural Networks via Meta Learning*. International Conference on Learning Representations.
- [105] Hao Yuan, Haiyang Yu, Shurui Gui, et al. (2020). *Explainability in Graph Neural Networks: A Taxonomic Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [106] Ben Finkelshtein, Xingyue Huang, Michael M. Bronstein, et al. (2023). *Cooperative Graph Neural Networks*. International Conference on Machine Learning.
- [107] Ana Lucic, Maartje ter Hoeve, Gabriele Tolomei, et al. (2021). *CF-GNNE explainer: Counterfactual Explanations for Graph Neural Networks*. International Conference on Artificial Intelligence and Statistics.
- [108] Xin Zheng, Yixin Liu, Shirui Pan, et al. (2022). *Graph Neural Networks for Graphs with Heterophily: A Survey*. arXiv.org.
- [109] Enyan Dai, M. Lin, Xiang Zhang, et al. (2023). *Unnoticeable Backdoor Attacks on Graph Neural Networks*. The Web Conference.

- [110] G. Jin, Yuxuan Liang, Yuchen Fang, et al. (2023). *Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey*. IEEE Transactions on Knowledge and Data Engineering.
- [111] Rex Ying, Ruining He, Kaifeng Chen, et al. (2018). *Graph Convolutional Neural Networks for Web-Scale Recommender Systems*. Knowledge Discovery and Data Mining.
- [112] Ziniu Hu, Yuxiao Dong, Kuansan Wang, et al. (2020). *GPT-GNN: Generative Pre-Training of Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [113] Sitao Luan, Chenqing Hua, Qincheng Lu, et al. (2022). *Revisiting Heterophily For Graph Neural Networks*. Neural Information Processing Systems.
- [114] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann (2018). *Predict then Propagate: Graph Neural Networks meet Personalized PageRank*. International Conference on Learning Representations.
- [115] Deli Chen, Yankai Lin, Wei Li, et al. (2019). *Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View*. AAAI Conference on Artificial Intelligence.
- [116] Yu Wang, Yuying Zhao, Yushun Dong, et al. (2022). *Improving Fairness in Graph Neural Networks via Mitigating Sensitive Attribute Leakage*. Knowledge Discovery and Data Mining.
- [117] Kaixiong Zhou, Xiao Huang, D. Zha, et al. (2021). *Dirichlet Energy Constrained Learning for Deep Graph Neural Networks*. Neural Information Processing Systems.

- [118] S. Jegelka (2022). *Theory of Graph Neural Networks: Representation and Learning*. arXiv.org.
- [119] Wei Jin, Lingxiao Zhao, Shichang Zhang, et al. (2021). *Graph Condensation for Graph Neural Networks*. International Conference on Learning Representations.
- [120] Simon Geisler, Tobias Schmidt, Hakan cSirin, et al. (2021). *Robustness of Graph Neural Networks at Scale*. Neural Information Processing Systems.
- [121] Zonghan Wu, Shirui Pan, Fengwen Chen, et al. (2019). *A Comprehensive Survey on Graph Neural Networks*. IEEE Transactions on Neural Networks and Learning Systems.
- [122] Keyulu Xu, Weihua Hu, J. Leskovec, et al. (2018). *How Powerful are Graph Neural Networks?*. International Conference on Learning Representations.
- [123] Jie Zhou, Ganqu Cui, Zhengyan Zhang, et al. (2018). *Graph Neural Networks: A Review of Methods and Applications*. AI Open.
- [124] Simon L. Batzner, Albert Musaelian, Lixin Sun, et al. (2021). *$E(3)$ -equivariant graph neural networks for data-efficient and accurate interatomic potentials*. Nature Communications.
- [125] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, et al. (2019). *SuperGlue: Learning Feature Matching With Graph Neural Networks*. Computer Vision and Pattern Recognition.
- [126] Zonghan Wu, Shirui Pan, Guodong Long, et al. (2020). *Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks*. Knowledge Discovery and Data Mining.

- [127] Shu Wu, Yuyuan Tang, Yanqiao Zhu, et al. (2018). *Session-based Recommendation with Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [128] Jiong Zhu, Yujun Yan, Lingxiao Zhao, et al. (2020). *Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs*. Neural Information Processing Systems.
- [129] Minjie Wang, Da Zheng, Zihao Ye, et al. (2019). *Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks..* Unpublished manuscript.
- [130] Mengzhang Li, and Zhanxing Zhu (2020). *Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting*. AAAI Conference on Artificial Intelligence.
- [131] Victor Garcia Satorras, and Joan Bruna (2017). *Few-Shot Learning with Graph Neural Networks*. International Conference on Learning Representations.
- [132] Yaqin Zhou, Shangqing Liu, J. Siow, et al. (2019). *Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks*. Neural Information Processing Systems.
- [133] Kenta Oono, and Taiji Suzuki (2019). *Graph Neural Networks Exponentially Lose Expressive Power for Node Classification*. International Conference on Learning Representations.
- [134] Lei Shi, Yifan Zhang, Jian Cheng, et al. (2019). *Skeleton-Based Action Recognition With Directed Graph Neural Networks*. Computer Vision and Pattern Recognition.
- [135] Zhanghao Wu, Paras Jain, Matthew A. Wright, et al. (2022).

- Representing Long-Range Context for Graph Neural Networks with Global Attention*. Neural Information Processing Systems.
- [136] Ziyang Wang, Wei Wei, G. Cong, et al. (2020). *Global Context Enhanced Graph Neural Networks for Session-based Recommendation*. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
- [137] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, et al. (2019). *Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems*. Knowledge Discovery and Data Mining.
- [138] Peixiang Zhong, Di Wang, and C. Miao (2019). *EEG-Based Emotion Recognition Using Regularized Graph Neural Networks*. IEEE Transactions on Affective Computing.
- [139] Tianxiang Zhao, Xiang Zhang, and Suhang Wang (2021). *GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks*. Web Search and Data Mining.
- [140] Qingsong Lv, Ming Ding, Qiang Liu, et al. (2021). *Are we really making much progress?: Revisiting, benchmarking and refining heterogeneous graph neural networks*. Knowledge Discovery and Data Mining.
- [141] Yue Yu, Jie Chen, Tian Gao, et al. (2019). *DAG-GNN: DAG Structure Learning with Graph Neural Networks*. International Conference on Machine Learning.
- [142] Jianheng Tang, Jiajin Li, Zi-Chao Gao, et al. (2022). *Rethinking Graph Neural Networks for Anomaly Detection*. International Conference on Machine Learning.

- [143] Jianan Zhao, Xiao Wang, C. Shi, et al. (2021). *Heterogeneous Graph Structure Learning for Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [144] R. Keisler (2022). *Forecasting Global Weather with Graph Neural Networks*. arXiv.org.
- [145] Maosen Li, Siheng Chen, Yangheng Zhao, et al. (2020). *Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction*. Computer Vision and Pattern Recognition.
- [146] Lingfei Wu, P. Cui, Jian Pei, et al. (2022). *Graph Neural Networks: Foundation, Frontiers and Applications*. Knowledge Discovery and Data Mining.
- [147] Zhenguang Liu, Peng Qian, Xiaoyang Wang, et al. (2021). *Combining Graph Neural Networks With Expert Knowledge for Smart Contract Vulnerability Detection*. IEEE Transactions on Knowledge and Data Engineering.
- [148] Hengrui Zhang, Qitian Wu, Junchi Yan, et al. (2021). *From Canonical Correlation Analysis to Self-supervised Graph Neural Networks*. Neural Information Processing Systems.
- [149] Yifei Shen, Yuanming Shi, Jun Zhang, et al. (2020). *Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis*. IEEE Journal on Selected Areas in Communications.
- [150] Yufeng Zhang, Xueli Yu, Zeyu Cui, et al. (2020). *Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks*. Annual Meeting of the Association for Computational Linguistics.

- [151] Qian Huang, Horace He, Abhay Singh, et al. (2020). *Combining Label Propagation and Simple Models Out-performs Graph Neural Networks*. International Conference on Learning Representations.
- [152] S. Schaefer, Daniel Gehrig, and D. Scaramuzza (2022). *AEGNN: Asynchronous Event-based Graph Neural Networks*. Computer Vision and Pattern Recognition.
- [153] Tianwen Chen, and R. C. Wong (2020). *Handling Information Loss of Graph Neural Networks for Session-based Recommendation*. Knowledge Discovery and Data Mining.
- [154] Yifei Shen, Jun Zhang, Shenghui Song, et al. (2022). *Graph Neural Networks for Wireless Communications: From Theory to Practice*. IEEE Transactions on Wireless Communications.
- [155] Kartik Sharma, Yeon-Chang Lee, S. Nambi, et al. (2022). *A Survey of Graph Neural Networks for Social Recommender Systems*. ACM Computing Surveys.
- [156] Tianlong Chen, Yongduo Sui, Xuxi Chen, et al. (2021). *A Unified Lottery Ticket Hypothesis for Graph Neural Networks*. International Conference on Machine Learning.
- [157] Cen Chen, Kenli Li, Wei Wei, et al. (2022). *Hierarchical Graph Neural Networks for Few-Shot Learning*. IEEE transactions on circuits and systems for video technology (Print).
- [158] Jiachen Li, Siheng Chen, Xiaoyong Pan, et al. (2022). *Cell clustering for spatial transcriptomics data with graph neural networks*. Nature Computational Science.

- [159] Seongjun Yun, Seoyoon Kim, Junhyun Lee, et al. (2022). *Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction*. Neural Information Processing Systems.
- [160] Asiri Wijesinghe, and Qing Wang (2022). *A New Perspective on "How Graph Neural Networks Go Beyond Weisfeiler-Lehman?"*. International Conference on Learning Representations.
- [161] Andrea Cini, Ivan Marisca, and C. Alippi (2021). *Filling the Gaps: Multivariate Time Series Imputation by Graph Neural Networks*. International
- [162] Lingfei Wu, Yu Chen, Kai Shen, et al. (2023). *Graph Neural Networks for Natural Language Processing: A Survey*. Found. Trends Mach. Learn..
- [163] Tianfu Li, Zheng Zhou, Sinan Li, et al. (2022). *The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study*. Mechanical systems and signal processing.
- [164] Petar Velickovic (2023). *Everything is Connected: Graph Neural Networks*. Current Opinion in Structural Biology.
- [165] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, et al. (2020). *Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models*. Journal of Cheminformatics.
- [166] Xiangguo Sun, Hongtao Cheng, Jia Li, et al. (2023). *All in One: Multi-Task Prompting for Graph Neural Networks*. Knowledge Discovery and Data Mining.

- [167] Xiao-Meng Zhang, Li Liang, Lin Liu, et al. (2021). *Graph Neural Networks and Their Current Applications in Bioinformatics*. Frontiers in Genetics.
- [168] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, et al. (2020). *Scaling Graph Neural Networks with Approximate PageRank*. Knowledge Discovery and Data Mining.
- [169] Jun Xia, Chengshuai Zhao, Bozhen Hu, et al. (2023). *Mole-BERT: Rethinking Pre-training Graph Neural Networks for Molecules*. International Conference on Learning Representations.
- [170] Saeed Rahmani, Asiye Baghbani, N. Bouguila, et al. (2023). *Graph Neural Networks for Intelligent Transportation Systems: A Survey*. IEEE transactions on intelligent transportation systems (Print).
- [171] Hao Chen, Yuan-Qi Bei, Qijie Shen, et al. (2024). *Macro Graph Neural Networks for Online Billion-Scale Recommender Systems*. The Web Conference.
- [172] Wenlong Liao, B. Bak-Jensen, J. Pillai, et al. (2021). *A Review of Graph Neural Networks and Their Applications in Power Systems*. Journal of Modern Power Systems and Clean Energy.
- [173] David Hin, Andrey Kan, Huaming Chen, et al. (2022). *LineVD: Statement-level Vulnerability Detection using Graph Neural Networks*. IEEE Working Conference on Mining Software Repositories.
- [174] Anton Tsitsulin, John Palowitch, Bryan Perozzi, et al. (2020). *Graph Clustering with Graph Neural Networks*. Journal of machine learning research.

- [175] Victor Fung, Jiaxin Zhang, Eric Juarez, et al. (2021). *Benchmarking graph neural networks for materials chemistry*. npj Computational Materials.
- [176] Yongxin Wang, Kris Kitani, and Xinshuo Weng (2021). *Joint Object Detection and Multi-Object Tracking with Graph Neural Networks*. IEEE International Conference on Robotics and Automation.
- [177] Danqing Wang, Pengfei Liu, Y. Zheng, et al. (2020). *Heterogeneous Graph Neural Networks for Extractive Document Summarization*. Annual Meeting of the Association for Computational Linguistics.
- [178] Kanchan Jha, S. Saha, and Hiteshi Singh (2022). *Prediction of protein-protein interaction using graph neural networks*. Scientific Reports.
- [179] M. Schuetz, J. K. Brubaker, and H. Katzgraber (2021). *Combinatorial optimization with physics-inspired graph neural networks*. Nature Machine Intelligence.
- [180] Meng Shen, Jinpeng Zhang, Liehuang Zhu, et al. (2021). *Accurate Decentralized Application Identification via Encrypted Traffic Analysis Using Graph Neural Networks*. IEEE Transactions on Information Forensics and Security.
- [181] Deyu Bo, Chuan Shi, Lele Wang, et al. (2023). *Specformer: Spectral Graph Neural Networks Meet Transformers*. International Conference on Learning Representations.
- [182] Mengqi Zhang, Shu Wu, Xueli Yu, et al. (2021). *Dynamic Graph Neural Networks for Sequential Recommendation*. IEEE Transactions on Knowledge and Data Engineering.

- [183] Jianjun Wei, Yue Liu, Xin Huang, et al. (2024). *Self-Supervised Graph Neural Networks for Enhanced Feature Extraction in Heterogeneous Information Networks*. 2024 5th International Conference on Machine Learning and Computer Application (ICMLCA).
- [184] Feng Yu, Yanqiao Zhu, Q. Liu, et al. (2020). *TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation*. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
- [185] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, et al. (2021). *FedGraphNN: A Federated Learning System and Benchmark for Graph Neural Networks*. arXiv.org.
- [186] Yulei Wu, Hongning Dai, and Haina Tang (2021). *Graph Neural Networks for Anomaly Detection in Industrial Internet of Things*. IEEE Internet of Things Journal.
- [187] Miltiadis Kofinas, Boris Knyazev, Yan Zhang, et al. (2024). *Graph Neural Networks for Learning Equivariant Representations of Neural Networks*. International Conference on Learning Representations.
- [188] Shuangli Li, Jingbo Zhou, Tong Xu, et al. (2021). *Structure-aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity*. Knowledge Discovery and Data Mining.
- [189] M. Balcilar, G. Renton, P. Héroux, et al. (2021). *Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective*. International Conference on Learning Representations.

- [190] Muhan Zhang, Pan Li, Yinglong Xia, et al. (2020). *Labeling Trick: A Theory of Using Graph Neural Networks for Multi-Node Representation Learning*. Neural Information Processing Systems.
- [191] Tristan Bilot, Nour El Madhoun, K. A. Agha, et al. (2023). *Graph Neural Networks for Intrusion Detection: A Survey*. IEEE Access.
- [192] Qitian Wu, Yiting Chen, Chenxiao Yang, et al. (2023). *Energy-based Out-of-Distribution Detection for Graph Neural Networks*. International Conference on Learning Representations.
- [193] P. Huang, Han-Hung Lee, Hwann-Tzong Chen, et al. (2021). *Text-Guided Graph Neural Networks for Referring 3D Instance Segmentation*. AAAI Conference on Artificial Intelligence.
- [194] Susheel Suresh, Vinith Budde, Jennifer Neville, et al. (2021). *Breaking the Limit of Graph Neural Networks by Improving the Assortativity of Graphs with Local Mixing Patterns*. Knowledge Discovery and Data Mining.
- [195] R. Liu, and Han Yu (2022). *Federated Graph Neural Networks: Overview, Techniques, and Challenges*. IEEE Transactions on Neural Networks and Learning Systems.
- [196] Lijing Wang, A. Adiga, Jiangzhuo Chen, et al. (2022). *Causal-GNN: Causal-Based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting*. AAAI Conference on Artificial Intelligence.
- [197] Fan Zhou, and Chengtai Cao (2021). *Overcoming Catastrophic Forgetting in Graph Neural Networks with Experience Replay*. AAAI Conference on Artificial Intelligence.

- [198] Vasimuddin, Sanchit Misra, Guixiang Ma, et al. (2021). *Dist-GNN: Scalable Distributed Training for Large-Scale Graph Neural Networks*. International Conference for High Performance Computing, Networking, Storage and Analysis.
- [199] Moshe Eliasof, E. Haber, and Eran Treister (2021). *PDE-GCN: Novel Architectures for Graph Neural Networks Motivated by Partial Differential Equations*. Neural Information Processing Systems.
- [200] Kexin Huang, Ying Jin, E. Candès, et al. (2023). *Uncertainty Quantification over Graph with Conformalized Graph Neural Networks*. Neural Information Processing Systems.
- [201] Matthias Fey, J. E. Lenssen, F. Weichert, et al. (2021). *GNNAutoScale: Scalable and Expressive Graph Neural Networks via Historical Embeddings*. International Conference on Machine Learning.
- [202] Van-Anh Nguyen, D. Q. Nguyen, Van Nguyen, et al. (2021). *ReGVD: Revisiting Graph Neural Networks for Vulnerability Detection*. 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion).
- [203] Nouhaila Innan, Abhishek Sawaika, Ashim Dhor, et al. (2023). *Financial Fraud Detection using Quantum Graph Neural Networks*. Quantum Machine Intelligence.
- [204] Jia Guo, and Chenyang Yang (2022). *Learning Power Allocation for Multi-Cell-Multi-User Systems With Heterogeneous Graph Neural Networks*. IEEE Transactions on Wireless Communications.

- [205] M. Maurizi, Chao Gao, and F. Berto (2022). *Predicting stress, strain and deformation fields in materials and structures with graph neural networks*. Scientific Reports.
- [206] Zi Ye, Y. J. Kumar, G. O. Sing, et al. (2022). *A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs*. IEEE Access.
- [207] Zemin Liu, Trung-Kien Nguyen, and Yuan Fang (2021). *Tail-GNN: Tail-Node Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [208] Lun Du, Xiaozhou Shi, Qiang Fu, et al. (2021). *GBK-GNN: Gated Bi-Kernel Graph Neural Networks for Modeling Both Homophily and Heterophily*. The Web Conference.
- [209] Weizhi Xu, Jun Wu, Qiang Liu, et al. (2022). *Evidence-aware Fake News Detection with Graph Neural Networks*. The Web Conference.
- [210] Shaocong Wang, Yi Li, Dingchen Wang, et al. (2023). *Echo state graph neural networks with analogue random resistive memory arrays*. Nature Machine Intelligence.
- [211] Rui Bing, Guan Yuan, Mu Zhu, et al. (2022). *Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications*. Artificial Intelligence Review.
- [212] Ziyu Lyu, Yue Wu, Junjie Lai, et al. (2023). *Knowledge Enhanced Graph Neural Networks for Explainable Recommendation*. IEEE Transactions on Knowledge and Data Engineering.
- [213] Hao Peng, Ruitong Zhang, Yingdong Dou, et al. (2021). *Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks*. ACM Trans. Inf. Syst..

- [214] Ying Xia, Chun-Qiu Xia, Xiaoyong Pan, et al. (2021). *Graph-Bind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues*. Nucleic Acids Research.
- [215] Aosong Feng, Chenyu You, Shiqiang Wang, et al. (2022). *KerGNNs: Interpretable Graph Neural Networks with Graph Kernels*. AAAI Conference on Artificial Intelligence.
- [216] Unknown Authors (2022). *Graph Neural Networks: Foundations, Frontiers, and Applications*. Unpublished manuscript.
- [217] Sitao Luan, Chenqing Hua, Qincheng Lu, et al. (2021). *Is Heterophily A Real Nightmare For Graph Neural Networks To Do Node Classification?*. arXiv.org.
- [218] Lilapati Waikhom, and Ripon Patgiri (2022). *A survey of graph neural networks in various learning paradigms: methods, applications, and challenges*. Artificial Intelligence Review.
- [219] Siyi Tang, Jared A. Dunnmon, Khaled Saab, et al. (2021). *Self-Supervised Graph Neural Networks for Improved Electroencephalographic Seizure Analysis*. International Conference on Learning Representations.
- [220] Veronika Thost, and Jie Chen (2021). *Directed Acyclic Graph Neural Networks*. International Conference on Learning Representations.
- [221] Ziwei Chai, Siqi You, Yang Yang, et al. (2022). *Can Abnormality be Detected by Graph Neural Networks?*. International Joint Conference on Artificial Intelligence.
- [222] Chengcheng Sun, Chenhao Li, Xiang Lin, et al. (2023).

Attention-based graph neural networks: a survey. Artificial Intelligence Review.

- [223] Mengqi Zhang, Shu Wu, Meng Gao, et al. (2022). *Personalized Graph Neural Networks With Attention Mechanism for Session-Aware Recommendation*. IEEE Transactions on Knowledge and Data Engineering.
- [224] Sai Munikoti, D. Agarwal, L. Das, et al. (2022). *Challenges and Opportunities in Deep Reinforcement Learning With Graph Neural Networks: A Comprehensive Review of Algorithms and Applications*. IEEE Transactions on Neural Networks and Learning Systems.
- [225] Ting-Li Huoh, Yan Luo, Peilong Li, et al. (2023). *Flow-Based Encrypted Network Traffic Classification With Graph Neural Networks*. IEEE Transactions on Network and Service Management.
- [226] Jiaqi Han, Yu Rong, Tingyang Xu, et al. (2022). *Geometrically Equivariant Graph Neural Networks: A Survey*. arXiv.org.
- [227] Hwan Kim, Byung Suk Lee, Won-Yong Shin, et al. (2022). *Graph Anomaly Detection With Graph Neural Networks: Current Status and Challenges*. IEEE Access.
- [228] Zeyang Zhang, Xin Wang, Ziwei Zhang, et al. (2022). *Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift*. Neural Information Processing Systems.
- [229] Yang Zhou, Jiuhong Xiao, Yuee Zhou, et al. (2022). *Multi-Robot Collaborative Perception With Graph Neural Networks*. IEEE Robotics and Automation Letters.
- [230] Xinyi Wu, A. Ajorlou, Zihui Wu, et al. (2023). *Demystifying*

- Oversmoothing in Attention-Based Graph Neural Networks*. Neural Information Processing Systems.
- [231] Yahui Long, Min Wu, Yong Liu, et al. (2022). *Pre-training graph neural networks for link prediction in biomedical networks*. Bioinform..
- [232] Andrea Cini, Ivan Marisca, F. Bianchi, et al. (2022). *Scalable Spatiotemporal Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [233] Zhen Zhang, Mohammed Haroon Dupty, Fan Wu, et al. (2023). *Factor Graph Neural Networks*. Journal of machine learning research.
- [234] Jianxin Chang, Chen Gao, Xiangnan He, et al. (2023). *Bundle Recommendation and Generation With Graph Neural Networks*. IEEE Transactions on Knowledge and Data Engineering.
- [235] J. Wang (2023). *A survey on graph neural networks*. EAI Endorsed Trans. e Learn..
- [236] Xusheng Zhao, Jia Wu, Hao Peng, et al. (2022). *Deep Reinforcement Learning Guided Graph Neural Networks for Brain Network Analysis*. Neural Networks.
- [237] Zahraa Al Sahili, and M. Awad (2023). *Spatio-Temporal Graph Neural Networks: A Survey*. arXiv.org.
- [238] R. Levie (2023). *A graphon-signal analysis of graph neural networks*. Neural Information Processing Systems.
- [239] Pengcheng Wang, Linping Tao, Mingwei Tang, et al. (2024). *Incorporating syntax and semantics with dual graph neural net-*

works for aspect-level sentiment analysis. Engineering applications of artificial intelligence.

- [240] Hu Dong, Longjie Li, Dongwen Tian, et al. (2024). *Dynamic link prediction by learning the representation of node-pair via graph neural networks*. Expert systems with applications.
- [241] Pengju Zhao, Wenjie Liao, Yuli Huang, et al. (2024). *Beam layout design of shear wall structures based on graph neural networks*. Automation in Construction.
- [242] Ming Chen, Yajian Jiang, Xiujuan Lei, et al. (2024). *Drug-Target Interactions Prediction Based on Signed Heterogeneous Graph Neural Networks*. Chinese journal of electronics.
- [243] P. Foroutan, and Salim Lahmiri (2024). *Deep Learning-Based Spatial-Temporal Graph Neural Networks for Price Movement Classification in Crude Oil and Precious Metal Markets*. Machine Learning with Applications.
- [244] Brook Wander, Muhammed Shuaibi, John R. Kitchen, et al. (2024). *CatTSunami: Accelerating Transition State Energy Calculations with Pretrained Graph Neural Networks*. ACS Catalysis.
- [245] Duantengchuan Li, Yuxuan Gao, Zhihao Wang, et al. (2024). *Homogeneous graph neural networks for third-party library recommendation*. Information Processing Management.
- [246] Yifan Duan, Guibin Zhang, Shilong Wang, et al. (2024). *CaT-GNN: Enhancing Credit Card Fraud Detection via Causal Temporal Graph Neural Networks*. arXiv.org.
- [247] R. Praveen, Aktalina Torogeldieva, B. Saravanan, et al. (2024). *Enhancing Intellectual Property Rights(IPR) Transparency with*

- Blockchain and Dual Graph Neural Networks*. 2024 First International Conference on Software, Systems and Information Technology (SSITCON).
- [248] Huiwei Wang, Tianhua Liu, Ziyu Sheng, et al. (2024). *Explanatory subgraph attacks against Graph Neural Networks*. Neural Networks.
- [249] Baoyu Jing, Dawei Zhou, Kan Ren, et al. (2024). *Causality-Aware Spatiotemporal Graph Neural Networks for Spatiotemporal Time Series Imputation*. International Conference on Information and Knowledge Management.
- [250] Zhongjian Zhang, Xiao Wang, Huichi Zhou, et al. (2024). *Can Large Language Models Improve the Adversarial Robustness of Graph Neural Networks?*. Knowledge Discovery and Data Mining.
- [251] Charilaos I. Kanatsoulis, and Alejandro Ribeiro (2024). *Counting Graph Substructures with Graph Neural Networks*. International Conference on Learning Representations.
- [252] Rajat Mishra, and S. Shridevi (2024). *Knowledge graph driven medicine recommendation system using graph neural networks on longitudinal medical records*. Scientific Reports.
- [253] Zhenyao Fang, and Qimin Yan (2024). *Towards accurate prediction of configurational disorder properties in materials using graph neural networks*. npj Computational Materials.
- [254] Jintu Zhang, Luigi Bonati, Enrico Trizio, et al. (2024). *Descriptor-Free Collective Variables from Geometric Graph Neural Networks..* Journal of Chemical Theory and Computation.

- [255] Nan Yin, Mengzhu Wang, Li Shen, et al. (2024). *Continuous Spiking Graph Neural Networks*. arXiv.org.
- [256] Liuxi Yan, and Yaoqun Xu (2024). *XGBoost-Enhanced Graph Neural Networks: A New Architecture for Heterogeneous Tabular Data*. Applied Sciences.
- [257] Xu Shen, P. Lió, Lintao Yang, et al. (2024). *Graph Rewiring and Preprocessing for Graph Neural Networks Based on Effective Resistance*. IEEE Transactions on Knowledge and Data Engineering.
- [258] S. K. Manivannan, Venkatesh Kavididevi, D. Muthukumaran, et al. (2024). *Graph Neural Networks for Resource Allocation Optimization in Healthcare Industry*. 2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (IC-SSAS).
- [259] Xingyang He (2024). *Graph neural networks in recommender systems*. Applied and Computational Engineering.
- [260] Zhe Zhao, Pengkun Wang, Haibin Wen, et al. (2024). *A Twist for Graph Classification: Optimizing Causal Information Flow in Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [261] Yafeng Yan, Shuyao He, Zhou Yu, et al. (2024). *Investigation of Customized Medical Decision Algorithms Utilizing Graph Neural Networks*. 2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE).
- [262] Zaishuo Xia, Han Yang, Binghui Wang, et al. (2024). *GNNCert: Deterministic Certification of Graph Neural Networks against*

Adversarial Perturbations. International Conference on Learning Representations.

- [263] Yicheng Zhou, P. Wang, Hao Dong, et al. (2024). *Make Graph Neural Networks Great Again: A Generic Integration Paradigm of Topology-Free Patterns for Traffic Speed Prediction*. International Joint Conference on Artificial Intelligence.
- [264] Shengyao Lu, Keith G. Mills, Jiao He, et al. (2024). *GOAt: Explaining Graph Neural Networks via Graph Output Attribution*. International Conference on Learning Representations.
- [265] Zhiyang Wang, J. Cerviño, and Alejandro Ribeiro (2024). *A Manifold Perspective on the Statistical Generalization of Graph Neural Networks*. arXiv.org.
- [266] Dilong Li, Chenghui Lu, Zi-xing Chen, et al. (2024). *Graph Neural Networks in Point Clouds: A Survey*. Remote Sensing.
- [267] A. Castro-Ospina, M. Solarte-Sanchez, L. Vega-Escobar, et al. (2024). *Graph-Based Audio Classification Using Pre-Trained Models and Graph Neural Networks*. Italian National Conference on Sensors.
- [268] Tianyi Zhao, Jian Kang, and Lu Cheng (2024). *Conformalized Link Prediction on Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [269] Wei Duan, Jie Lu, Yu Guang Wang, et al. (2024). *Layer-diverse Negative Sampling for Graph Neural Networks*. Trans. Mach. Learn. Res..
- [270] Xuexiong Luo, Jia Wu, Jian Yang, et al. (2024). *Graph Neural Networks for Brain Graph Learning: A Survey*. International Joint Conference on Artificial Intelligence.

- [271] Alessandro De Carlo, D. Ronchi, Marco Piastra, et al. (2024). *Predicting ADMET Properties from Molecule SMILE: A Bottom-Up Approach Using Attention-Based Graph Neural Networks*. Pharmaceutics.
- [272] Sahab Zandi, Kamesh Korangi, Mar'ia 'Oskarsd'ottir, et al. (2024). *Attention-based Dynamic Multilayer Graph Neural Networks for Loan Default Prediction*. European Journal of Operational Research.
- [273] Haihong Zhao, Bo Yang, Jiaxu Cui, et al. (2024). *Effective Fault Scenario Identification for Communication Networks via Knowledge-Enhanced Graph Neural Networks*. IEEE Transactions on Mobile Computing.
- [274] Rufan Yao, Zhenhua Shen, Xinyi Xu, et al. (2024). *Knowledge mapping of graph neural networks for drug discovery: a bibliometric and visualized analysis*. Frontiers in Pharmacology.
- [275] Tuan Vinh, Loc Nguyen, Quang H. Trinh, et al. (2024). *Predicting Cardiotoxicity of Molecules Using Attention-Based Graph Neural Networks*. Journal of Chemical Information and Modeling.
- [276] Inaam Ashraf, Janine Strother, L. Hermes, et al. (2024). *Physics-Informed Graph Neural Networks for Water Distribution Systems*. AAAI Conference on Artificial Intelligence.
- [277] Zachary Smith, Michael Strobel, Bodhi P. Vani, et al. (2024). *Graph Attention Site Prediction (GrASP): Identifying Drug-gable Binding Sites Using Graph Neural Networks with Attention*. Journal of Chemical Information and Modeling.

- [278] S. Abadal, Pablo Galván, Alberto Mármol, et al. (2024). *Graph neural networks for electroencephalogram analysis: Alzheimer’s disease and epilepsy use cases*. Neural Networks.
- [279] Maximilian Pflueger, David J. Tena Cucala, and Egor V. Kostylev (2024). *Recurrent Graph Neural Networks and Their Connections to Bisimulation and Logic*. AAAI Conference on Artificial Intelligence.
- [280] H. Mohammadi, and Waldemar Karwowski (2024). *Graph Neural Networks in Brain Connectivity Studies: Methods, Challenges, and Future Directions*. Brain Science.
- [281] Yongduo Sui, Xiang Wang, Tianlong Chen, et al. (2024). *Inductive Lottery Ticket Learning for Graph Neural Networks*. Journal of Computational Science and Technology.
- [282] Jie Peng, Runlin Lei, and Zhewei Wei (2024). *Beyond Over-smoothing: Uncovering the Trainability Challenges in Deep Graph Neural Networks*. International Conference on Information and Knowledge Management.
- [283] Shan Zhao, Ioannis Prapas, Ilektra Karasante, et al. (2024). *Causal Graph Neural Networks for Wildfire Danger Prediction*. arXiv.org.
- [284] M. A. Nabian (2024). *X-MeshGraphNet: Scalable Multi-Scale Graph Neural Networks for Physics Simulation*. arXiv.org.
- [285] Jiacheng Cen, Anyi Li, Ning Lin, et al. (2024). *Are High-Degree Representations Really Unnecessary in Equivariant Graph Neural Networks?*. Neural Information Processing Systems.
- [286] Yachao Yang, Yanfeng Sun, Shaofan Wang, et al. (2024). *Graph*

Neural Networks with Soft Association between Topology and Attribute. AAAI Conference on Artificial Intelligence.

- [287] Youjia Li, Vishu Gupta, Muhammed Nur Talha Kilic, et al. (2024). *Hybrid-LLM-GNN: Integrating Large Language Models and Graph Neural Networks for Enhanced Materials Property Prediction.* Digital Discovery.
- [288] Zhenbei Guo, Fuliang Li, Jiaying Shen, et al. (2024). *ConfigReco: Network Configuration Recommendation With Graph Neural Networks.* IEEE Network.
- [289] A. Gnanabaskaran, K. Bharathi, S. P. Nandakumar, et al. (2024). *Enhanced Drug-Drug Interaction Prediction with Graph Neural Networks and SVM.* 2024 International Conference on Knowledge Engineering and Communication Systems (ICK-ECS).
- [290] Beibei Wang, Bo Jiang, and Chris H. Q. Ding (2024). *FL-GNNs: Robust Network Representation via Feature Learning Guided Graph Neural Networks.* IEEE Transactions on Network Science and Engineering.
- [291] Daniel Abode, Ramoni O. Adeogun, and Gilberto Berardinelli (2024). *Power Control for 6G In-Factory Subnetworks With Partial Channel Information Using Graph Neural Networks.* IEEE Open Journal of the Communications Society.
- [292] Tianxiang Zhao, Xiang Zhang, and Suhan Wang (2024). *Disambiguated Node Classification with Graph Neural Networks.* The Web Conference.
- [293] Christian Hausleitner, Heimo Mueller, Andreas Holzinger, et al. (2024). *Collaborative weighting in federated graph neural net-*

works for disease classification with the human-in-the-loop. Scientific Reports.

- [294] Shan Zhao, Zhaiyu Chen, Zhitong Xiong, et al. (2024). *Beyond Grid Data: Exploring graph neural networks for Earth observation.* IEEE Geoscience and Remote Sensing Magazine.
- [295] T. Konstantin Rusch, Nathan Kirk, M. Bronstein, et al. (2024). *Message-Passing Monte Carlo: Generating low-discrepancy point sets via graph neural networks.* Proceedings of the National Academy of Sciences of the United States of America.
- [296] Fali Wang, Tianxiang Zhao, and Suhan Wang (2024). *Distribution Consistency based Self-Training for Graph Neural Networks with Sparse Labels.* Web Search and Data Mining.
- [297] Bingyao Liu, Iris Li, Jianhua Yao, et al. (2024). *Unveiling the Potential of Graph Neural Networks in SME Credit Risk Assessment.* 2024 5th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI).
- [298] Zhenyao Fang, and Qimin Yan (2024). *Leveraging Persistent Homology Features for Accurate Defect Formation Energy Predictions via Graph Neural Networks.* Chemistry of Materials.
- [299] Michael Benedikt, Chia-Hsuan Lu, Boris Motik, et al. (2024). *Decidability of Graph Neural Networks via Logical Characterizations.* International Colloquium on Automata, Languages and Programming.
- [300] Yuelin Zhang, Jiacheng Cen, Jiaqi Han, et al. (2024). *Improving Equivariant Graph Neural Networks on Large Geometric Graphs via Virtual Nodes Learning.* International Conference on Machine Learning.

- [301] Caterina Graziani, Tamara Drucks, Fabian Jögl, et al. (2024). *The Expressive Power of Path-Based Graph Neural Networks*. International Conference on Machine Learning.
- [302] Dai Shi, Andi Han, Lequan Lin, et al. (2024). *Design Your Own Universe: A Physics-Informed Agnostic Method for Enhancing Graph Neural Networks*. International Journal of Machine Learning and Cybernetics.
- [303] Jiang Yuan, Shanxiong Chen, Bofeng Mo, et al. (2024). *R-GNN: recurrent graph neural networks for font classification of oracle bone inscriptions*. Heritage Science.
- [304] Haitao Wang, Zelin Liu, Mingjun Li, et al. (2024). *A Gearbox Fault Diagnosis Method Based on Graph Neural Networks and Markov Transform Fields*. IEEE Sensors Journal.
- [305] Ferial Abuhantash, Mohd Khalil Abu Hantash, and Aamna Al-Shehhi (2024). *Comorbidity-based framework for Alzheimer’s disease classification using graph neural networks*. Scientific Reports.
- [306] Yassine Abbahaddou, Sofiane Ennadir, J. Lutzeyer, et al. (2024). *Bounding the Expected Robustness of Graph Neural Networks Subject to Node Feature Attacks*. International Conference on Learning Representations.
- [307] Renhong Huang, Jiarong Xu, Xin Jiang, et al. (2024). *Measuring Task Similarity and Its Implication in Fine-Tuning Graph Neural Networks*. AAAI Conference on Artificial Intelligence.
- [308] Yue Jiang, Changkong Zhou, Vikas Garg, et al. (2024). *Graph4GUI: Graph Neural Networks for Representing Graphi-*

cal User Interfaces. International Conference on Human Factors in Computing Systems.

- [309] Bin Wang, Yadong Xu, Manyi Wang, et al. (2024). *Gear Fault Diagnosis Method Based on the Optimized Graph Neural Networks*. IEEE Transactions on Instrumentation and Measurement.
- [310] Thiago H. Silva, and Daniel Silver (2024). *Using graph neural networks to predict local culture*. Environment and Planning B Urban Analytics and City Science.
- [311] Xin Zhang, Zhen Xu, Yue Liu, et al. (2024). *Robust Graph Neural Networks for Stability Analysis in Dynamic Networks*. 2024 3rd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE).
- [312] Mengfang Sun, Wenying Sun, Ying Sun, et al. (2024). *Applying Hybrid Graph Neural Networks to Strengthen Credit Risk Analysis*. 2024 3rd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE).
- [313] Xin Zeng, Fan-Fang Meng, Meng-Liang Wen, et al. (2024). *GNNGL-PPI: multi-category prediction of protein-protein interactions using graph neural networks based on global graphs and local subgraphs*. BMC Genomics.
- [314] Ziang Chen, Xiaohan Chen, Jialin Liu, et al. (2024). *Expressive Power of Graph Neural Networks for (Mixed-Integer) Quadratic Programs*. arXiv.org.
- [315] Takaaki Fujita (2024). *Superhypergraph Neural Networks and Plithogenic Graph Neural Networks: Theoretical Foundations*. arXiv.org.

- [316] Mahdi Saleh, Michael Sommersperger, N. Navab, et al. (2024). *Physics-Encoded Graph Neural Networks for Deformation Prediction under Contact*. IEEE International Conference on Robotics and Automation.
- [317] Mohammed Aburidi, and Roummel F. Marcia (2024). *Topological Adversarial Attacks on Graph Neural Networks Via Projected Meta Learning*. IEEE Conference on Evolving and Adaptive Intelligent Systems.
- [318] Zhonghao Wang, Danyu Sun, Sheng Zhou, et al. (2024). *NoisyGL: A Comprehensive Benchmark for Graph Neural Networks under Label Noise*. Neural Information Processing Systems.
- [319] Rostislav Horčík, and Gustav Sír (2024). *Expressiveness of Graph Neural Networks in Planning Domains*. International Conference on Automated Planning and Scheduling.
- [320] Jianshan Sun, Suyuan Mei, Kun Yuan, et al. (2024). *Prerequisite-Enhanced Category-Aware Graph Neural Networks for Course Recommendation*. ACM Transactions on Knowledge Discovery from Data.
- [321] Langsha Li, Feng Qiang, and Li Ma (2024). *Advancing Cybersecurity: Graph Neural Networks in Threat Intelligence Knowledge Graphs*. International Conference on Algorithms, Software Engineering, and Network Security.
- [322] Renqiang Luo, Huafei Huang, Shuo Yu, et al. (2024). *FUGNN: Harmonizing Fairness and Utility in Graph Neural Networks*. Knowledge Discovery and Data Mining.

- [323] Shouheng Li, F. Geerts, Dongwoo Kim, et al. (2024). *Towards Bridging Generalization and Expressivity of Graph Neural Networks*. International Conference on Learning Representations.
- [324] Yidong Liao, Xiao-Ming Zhang, and Chris Ferrie (2024). *Graph Neural Networks on Quantum Computers*. arXiv.org.
- [325] Yufeng Wang, and Charith Mendis (2024). *TGLite: A Lightweight Programming Framework for Continuous-Time Temporal Graph Neural Networks*. International Conference on Architectural Support for Programming Languages and Operating Systems.
- [326] Ping Liu, Haichao Wei, Xiaochen Hou, et al. (2024). *LinkSAGE: Optimizing Job Matching Using Graph Neural Networks*. Knowledge Discovery and Data Mining.
- [327] Alan John Varghese, Zhen Zhang, and G. Karniadakis (2024). *SympGNNs: Symplectic Graph Neural Networks for identifying high-dimensional Hamiltonian systems and node classification*. Neural Networks.
- [328] Giuseppe Alessio D’Inverno, M. Bianchini, and F. Scarselli (2024). *VC dimension of Graph Neural Networks with Pfaffian activation functions*. Neural Networks.