INTERNATIONAL JOURNAL OF PURE SCIENCE RESEARCH STUDIES (IJPSRS)
(Open Access, Double-Blind Peer Reviewed Journal)
EDU SCHOOL
ijpsrs.com
ISSN Online: ISSN Print

Graph Neural Networks: Mathematical Foundations and Algorithmic Frameworks

Lejo J Manavalan

Assistant Professor and Research Guide, Department of Mathematics, Little Flower College, Guruvayur, India

Article Information

Received: 3rd February 2026
Received in revised form: 5th March 2026
Accepted: 7th April 2026
Available online: 14th May 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.5281/zenodo.20151052

Abstract

Graph neural networks (GNNs) extend deep learning to data supported on graph-structured domains. Motivated by problems in chemistry, social networks, recommender systems and relational reasoning, GNN architectures have matured into a rigorous subfield combining graph theory, spectral analysis and approximation theory. This paper offers a mathematical review of GNN foundations. We begin with spectral formulations based on the graph Laplacian, derive the message-passing neural network framework, and survey expressive-power results including the equivalence of 1-WL colour refinement and standard message-passing. Representative architectures GCN, GraphSAGE, GAT, and GIN are compared on citation-network benchmarks. The paper closes with a discussion of over-smoothing, long-range dependency, and recent higher-order extensions.

Keywords: Graph Neural Network, Message Passing, Spectral Graph Theory, Graph Convolution, Weisfeiler-Leman.

1. INTRODUCTION

Many datasets of contemporary scientific and commercial interest have intrinsic graph structure. Molecules are graphs with atoms as nodes and bonds as edges. Citation networks link papers by their references. Social graphs encode friendships, follower relations, and communication. Knowledge graphs encode entity-relation triples capturing world knowledge. Protein interaction networks, transportation networks, recommender interaction graphs, and program dependency graphs all share this relational structure. Conventional convolutional and recurrent deep-learning architectures exploit grid or sequence regularities shift-equivariance for images, temporal recurrence for sequences that are unavailable in general graph data. Graph neural networks (GNNs) generalise convolution to arbitrary graphs by exchanging information along edges, enabling feature learning on relational data.[1,2]

GNNs emerged from two converging lines of development. The first was spectral graph signal processing, which generalises Fourier analysis to functions on graphs via the graph Laplacian spectrum; the second was the early neural-network work on structured prediction (recursive neural networks, neural networks for graphs by Scarselli and colleagues in 2009).[3] The modern GNN era began with the work of Bruna and colleagues on spectral graph convolutions in 2013, the localised Chebyshev-polynomial approximation of Defferrard and colleagues in 2016, and the highly influential simplified graph convolutional network of Kipf and Welling in 2017.[6] Since then the field has expanded dramatically, supporting a principled theory of expressiveness, convergence, and generalisation, together with a wide practical toolkit and substantial industry adoption.

This paper surveys the mathematical foundations, dominant architectural frameworks, and open problems of graph neural networks. Section 2 reviews spectral graph theory and its relationship to graph convolution. Section 3 presents the message-passing framework that unifies most modern architectures. Section 4 surveys representative architectures. Section 5 addresses expressiveness results, particularly the equivalence of standard message-passing to the 1-dimensional Weisfeiler–Leman test. Section 6 presents empirical comparisons, and Sections 7–8 discuss challenges and conclusions.

2. GRAPH THEORY AND SPECTRAL FOUNDATIONS

Let G = (V, E) be an undirected graph with |V| = N vertices and |E| edges, adjacency matrix :

$$A \in \{0,1\}^{N\times N} A_{ij} = 1 \quad (1)$$

if ij ∈ E, else 0), degree matrix D = diag(d_1,…, d_N) where:

$$d_i = \sum_j A_{ij} \quad (2)$$

and combinatorial Laplacian L = D − A. The combinatorial Laplacian is symmetric positive semi-definite; its multiplicity of the zero eigenvalue equals the number of connected components, and its second-smallest eigenvalue (the Fiedler value) measures algebraic connectivity.

The normalised Laplacian $\hat{L} = I - D^{-1/2}AD^{-1/2}$ has eigenvalues in [0,2]; its eigenvectors $\{u_k\}$ and eigenvalues $\{\lambda_k\}$ provide a Fourier basis for graph signals $x \in \mathbb{R}^N$, with the graph Fourier transform $\hat{x} = U^T x$ and its inverse $x = U\hat{x}_3$. This construction generalises the classical Fourier transform: on the discrete circle Z/N with its cycle-graph Laplacian, the eigenvectors of L reduce to the classical discrete Fourier basis. Graph signal-processing concepts filtering, frequency bands, smoothness, multi-resolution analysis translate to arbitrary graphs via this spectral decomposition.
A graph convolution is defined by element-wise multiplication of spectral representations:

$$y = U g_\theta(\Lambda) U^T x \quad (3)$$

where U is the eigenbasis, Λ = diag(λ_1,…, λ_N) , and g_θ a learnable filter function. Direct spectral evaluation requires O(N³) eigendecomposition and is non-transferable across graphs of different size. Localised filters admit polynomial parametrisations:

$$g_\theta(\Lambda) = \sum_k \theta_k T_k(\tilde{\Lambda}) \quad (4)$$

where T_k are Chebyshev polynomials and $\tilde{\Lambda}$ is a rescaled Laplacian, yielding O(|E|) complexity per evaluation and K-hop spatial locality for a polynomial of degree K[6]. The first-order Chebyshev approximation, augmented with renormalisation, yields the popular Graph Convolutional Network (GCN) layer. ChebNet with higher-order polynomials enjoys stronger expressive power at the cost of slightly higher per-layer compute, and has proven effective on tasks where longer-range dependencies matter.

Spectral analysis also provides theoretical tools for studying GNN properties. The spectral radius of the aggregation operator governs the stability of deep architectures; spectral smoothness of signals captures homophily (the tendency of connected nodes to share labels); and the interplay between graph spectrum and filter design explains phenomena such as over-smoothing and over-squashing that emerge in deep GNNs. The spectral perspective is complementary to the spatial (message-passing) view developed in the following sections.

3. MESSAGE PASSING FRAMEWORK

Gilmer and colleagues in 2017 introduced the unifying message-passing neural network (MPNN) framework that subsumes most contemporary graph neural architectures.[4] At layer l, each node v updates its representation $h_v^{(l)}$ via messages:

$$m_v = AGG(\{M(h_v, h_u, e_{uv})|u \in N(v)\}) \quad (5)$$

followed by an update:

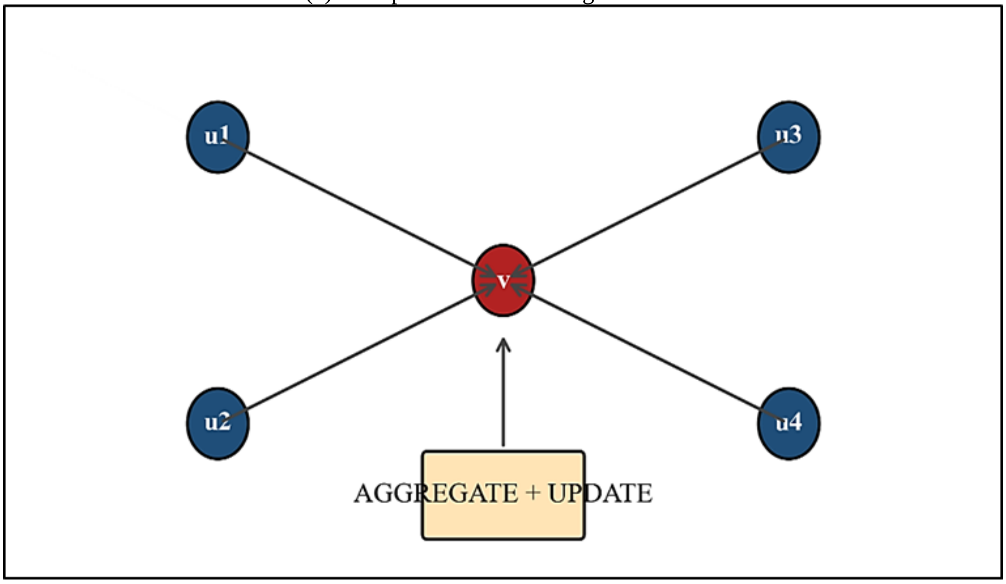
$$h_v^{(1+1)} = \dot{U}(h_v^{(1)}, m_v) \quad (6)$$

The aggregation AGG is a permutation-invariant function (sum, mean, max, attention-weighted, set-theoretic), ensuring that the output is independent of the arbitrary ordering of neighbours. M is a learnable message function (often a multi-layer perceptron or a linear transformation of concatenated features), and U is a learnable update function (typically a gated recurrent unit, multi-layer perceptron, or simple addition followed by nonlinearity).

The design of AGG, M, and U determines the expressive capacity and computational profile of the architecture. Sum aggregation, combined with injective update, yields maximally expressive message passing. Mean aggregation is appropriate when absolute neighbour count is irrelevant (common in citation networks). Max aggregation emphasises the most prominent neighbour feature. Attention-weighted aggregation adaptively weights neighbours according to learned relevance. Set-theoretic aggregators (DeepSets, PNA) combine multiple aggregation strategies to capture richer neighbourhood distributions. Beyond single-step aggregation, iterative message passing stacks L layers, giving each node access to L-hop neighbourhood information.

Readout operations convert node representations to graph-level predictions. Standard choices are sum-, mean-, or max-pooling over node features; attention-based pooling (Set2Set, GMP) and hierarchical pooling (DiffPool, Top-k pooling, SAGPool) enable richer graph-level representations. The full MPNN framework is shown schematically in Figure 1, with node v receiving messages from each neighbour, aggregating them, and updating its own representation.

Fig 1: Message passing: at each layer a node v aggregates transformed features from its neighbourhood N(v) and updates its embedding.[4]



4. REPRESENTATIVE ARCHITECTURES

The Graph Convolutional Network (GCN) of Kipf and Welling uses first-order spectral approximations, giving a layer-wise propagation rule:

$$H^{(1+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)}) \tag{7}$$

with:

$$\tilde{A} = \tilde{D}^{-1/2}(A + I)\tilde{D}^{-1/2} \tag{8}$$

where I is added to preserve self-information via self-loops.[6] This renormalisation trick stabilises training and provides a simple yet effective baseline that has become the de facto standard entry point for graph-based deep learning. The GCN propagation can be interpreted as smoothing signals over the graph Laplacian's low-frequency components, consistent with the homophily assumption underlying many benchmark graph tasks.

GraphSAGE (SAmple and aggreGatE), introduced by Hamilton and colleagues, generalises the GCN framework with support for inductive learning on large or evolving graphs.[7] GraphSAGE samples fixed-size neighbourhoods (typically 10–25 neighbours per node), applies aggregator functions (mean, pooling, or LSTM), and concatenates with self-features. Neighbour sampling bounds per-node computation, enabling training on graphs with billions of edges via minibatch gradient descent. GraphSAGE has been deployed in production recommender systems at major technology companies including Pinterest and Uber.

The Graph Attention Network (GAT) replaces uniform aggregation with attention coefficients learned between neighbours, providing adaptive weighting that emphasises informative neighbours.[8] Each neighbour's contribution is weighted by a learned attention score:

$$\alpha_{ij} = \mathrm{softmax}_j\left(\mathrm{LeakyReLU}\left(a^T\left[Wh_i \parallel Wh_j\right]\right)\right) \tag{9}$$

where W is a shared linear transformation and a is an attention vector. Multi-head attention averages or concatenates outputs from multiple parallel attention computations, further improving expressiveness. GAT is particularly effective on heterophilic graphs where connected nodes may carry different labels.

The Graph Isomorphism Network (GIN) uses sum aggregation with a multi-layer perceptron update:

$$h_v^{(l+1)} = \mathrm{MLP}^{(l)}\left((1 + \epsilon^{(l)})\, h_v^{(l)} + \sum\nolimits_{u \in N(v)} h_u^{(l)}\right) \tag{10}$$

Xu and colleagues showed that this formulation is as expressive as the 1-dimensional Weisfeiler–Leman graph-isomorphism test, the strongest possible for message-passing architectures[5]. GIN achieves strong performance on graph-level tasks including molecular property prediction on the ogbg-molhiv and QM9 benchmarks. Additional architectures include APPNP (personalised PageRank propagation), MixHop (mixing different hop neighbourhoods), GCNII (residual connections to avoid over-smoothing), and GraphTransformer (self-attention with structural encoding). Heterogeneous graph neural networks (R-GCN, HAN, HGT) handle graphs with multiple node and edge types, essential for knowledge graphs and multi-modal relational data.

5. EXPRESSIVE POWER AND WEISFEILER–LEMAN EQUIVALENCE

Xu and colleagues established a fundamental connection between message-passing GNNs and the 1-dimensional Weisfeiler–Leman (1-WL) graph-isomorphism test[5]. The 1-WL test iteratively refines node labels by hashing each node's current label together with the multiset of its neighbours' labels; two graphs are declared potentially isomorphic if their final label multisets match. Standard message-passing GNNs execute the same structural procedure with learned hash functions, and therefore cannot distinguish non-isomorphic graphs that 1-WL cannot distinguish. The Graph Isomorphism Network achieves the maximum expressiveness in this class under sum aggregation and injective update functions.

The 1-WL test is strictly weaker than graph isomorphism: classical counterexamples include pairs of k-regular graphs that are not isomorphic but share identical 1-WL refinements. To overcome this expressiveness ceiling, several higher-order architectures have been proposed.[9] k-WL inspired architectures operate on ordered k-tuples of nodes, attaining the stronger k-dimensional Weisfeiler–Leman test's expressive power at a computational cost exponential in k. Subgraph GNNs compute node representations by aggregating over multiple subgraphs rooted at each node, typically achieving expressive power between 2-WL and 3-WL while remaining computationally tractable. Simplicial and cell-complex networks generalise the message-passing framework to higher-dimensional algebraic-topological structures, naturally encoding triangles, cycles, and higher-order interactions. Graph transformers drop the strict locality constraint, using self-attention across all node pairs with structural positional encodings (Laplacian eigenvectors, random-walk features) to preserve graph structure information. Each of these approaches balances expressive power, computational cost, and empirical effectiveness differently; choice depends on application and scale.

6. EMPIRICAL COMPARISON

Figure 2 shows test accuracies of representative architectures on the Cora, Citeseer, and Pubmed citation benchmarks. Architectures with adaptive aggregation (GAT) and maximally expressive aggregation (GIN) tend to outperform simpler GCN on small graphs but converge on larger graphs. Personal-PageRank-based APPNP variants balance expressiveness with computational efficiency.[10] Table 1 summarises principal architectural tradeoffs.

Fig 2: Test accuracy on standard citation benchmarks for representative GNN architectures.[5,6,7,8,10]

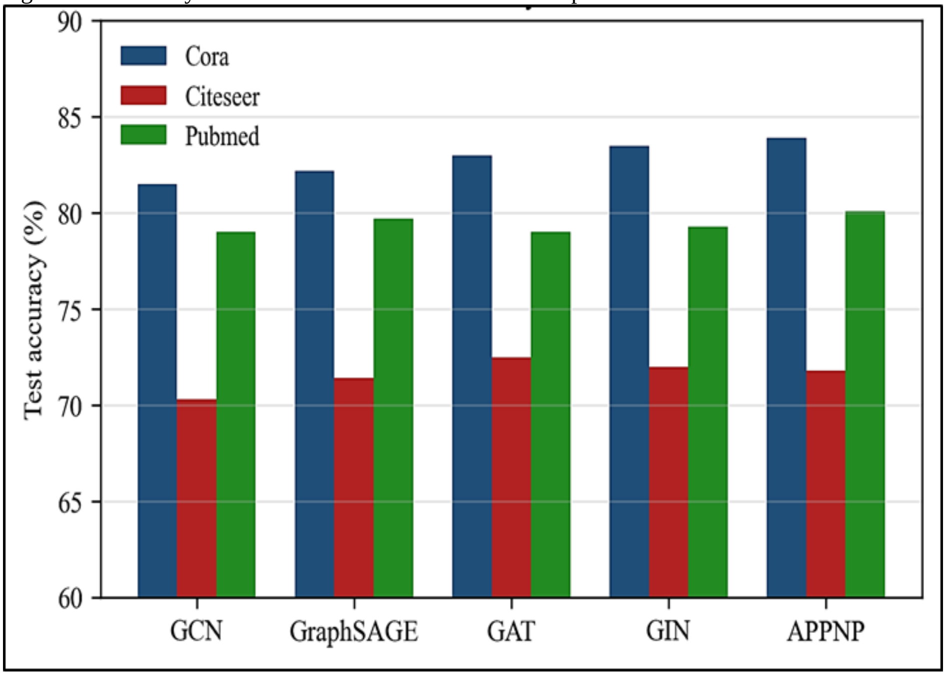


Table 1. Architectural comparison of representative GNNs. [5,6,7,8,10]

Architecture	Aggregation	Spectral/Spatial	Key property
GCN	Symmetric-normalised mean	Spectral (Chebyshev order 1)	Simple, scalable baseline
GraphSAGE	Mean / pool / LSTM	Spatial sampling	Inductive on large graphs
GAT	Attention-weighted sum	Spatial	Adaptive neighbour weighting
GIN	Injective sum + MLP	Spatial	1-WL expressive
APPNP	Personalised PageRank	Propagation	Decouples depth from expressiveness

7. CHALLENGES AND FUTURE DIRECTIONS

Several challenges continue to structure the research agenda. First, over-smoothing: deep GNNs collapse node representations as layers increase, so that nodes within a connected component converge to identical embeddings.[11] Mathematically, this occurs because repeated graph-convolution operations are analogous to repeatedly applying a low-pass filter, eventually projecting signals onto the Laplacian's null space. Residual connections (GCNII), skip connections, PairNorm normalisation, personalised-PageRank propagation (APPNP), and spectral regularisers mitigate over-smoothing, enabling effective networks of 20+ layers on some benchmarks.

Second, over-squashing and long-range dependency: message-passing restricts information flow to local neighbourhoods, so information from nodes many hops away is exponentially compressed through intermediate aggregation an effect termed 'over-squashing' by Topping and colleagues. Graph rewiring techniques, graph curvature-guided adjustment, and the insertion of virtual global nodes have been proposed as remedies. Graph transformers and higher-order architectures address long-range dependency by breaking the strict locality of message-passing,[9] though at higher computational cost and with less-developed theory.

Third, generalisation theory for GNNs on graphs drawn from complex distributions provides a growing mathematical foundation. The 'graphon' limit treating large graphs as samples from a continuous kernel enables transferability results establishing that GNNs trained on moderate-sized graphs generalise predictably to larger graphs of similar structure.[12] PAC-Bayesian bounds, Rademacher complexity analyses, and neural-tangent-kernel results are advancing the theoretical understanding of generalisation, overparameterisation, and implicit regularisation in GNNs.

Fourth, scalability remains an engineering challenge. Techniques including neighbour sampling, layer-wise sampling, historical embeddings (GraphSAINT, Cluster-GCN), distributed training across partitioned graphs, and GPU-optimised sparse operations have enabled training on graphs with billions of nodes. Frameworks such as DGL, PyTorch Geometric, and JAX-based Jraph continue to reduce the engineering cost of large-scale GNN deployment. Fifth, robustness and explainability: GNNs are sensitive to adversarial perturbations of graph structure, and explanations of GNN predictions require methods tailored to relational data (GNNExplainer, PGExplainer, subgraph-level interpretability). Research in these directions is expanding rapidly alongside the broader deep-learning interpretability literature.

8. APPLICATIONS ACROSS DOMAINS

GNN applications span diverse scientific and industrial domains. In chemistry and drug discovery, GNNs predict molecular properties (QM9, MoleculeNet benchmarks), generate molecular structures via graph generative models, and model chemical reactions. AlphaFold's initial protein-structure predictions used graph-attention components; AlphaFold-Multimer and more recent derivatives continue to integrate graph architectures. In materials science, GNNs predict formation energies, band gaps, and phonon spectra of crystalline materials, with dedicated crystal-graph architectures (CGCNN, MEGNet, ALIGNN) achieving state-of-the-art accuracy on materials datasets.

In recommender systems, GNNs represent user item interactions as bipartite graphs and leverage message passing for collaborative filtering. PinSage at Pinterest, LightGCN, and Uber's graph-based recommenders are production deployments operating on graphs with billions of edges. In social-network analysis, GNNs predict user influence, community structure, and content diffusion. In knowledge-graph completion, relational GCNs and knowledge-graph embedding methods predict missing triples in Freebase, WordNet, and domain-specific knowledge graphs.

In physics and engineering, GNNs model particle physics (jet tagging, event reconstruction at ATLAS and CMS), fluid dynamics (mesh-based GNN simulators such as DeepMind's MeshGraphNet), rigid-body dynamics, and power-grid analysis. In transportation and logistics, GNNs predict traffic flow, route congestion, and supply-chain disruption on road and transit networks. In finance, GNNs detect fraudulent transactions, estimate credit risk from transaction graphs, and model portfolio-risk dependencies.

9. CONCLUSION

Graph neural networks have matured into a principled methodology at the intersection of spectral graph theory, algebraic topology, and deep learning. Expressiveness, generalisation, and computational cost are each the subject of active theoretical progress, supported by a broad software ecosystem and large-scale empirical benchmarking. Empirical performance on citation, molecular, recommender, and knowledge-graph benchmarks continues to improve, and the mathematical tools developed for GNNs expressive-power hierarchies, spectral analysis, graphon limits are migrating into broader areas of relational machine learning. The field is now embedded in the broader geometric deep-learning programme, which systematically treats data with symmetry and structure across manifolds, grids, groups, graphs, and geodesics. Further progress will likely come from tighter integration with scientific domain knowledge (physics-informed architectures, causal inference), improved methods for heterogeneous and dynamic graphs, and robust foundations for large-scale deployment on graphs with billions of nodes.

REFERENCES

1. Bronstein MM, Bruna J, Cohen T, Veličković P. 2021. Geometric deep learning: grids, groups, graphs, geodesics, and gauges. arXiv. 2104.13478.
2. Hamilton WL. 2020. Graph representation learning. Synth Lect Artif Intell Mach Learn. 14(3):1–159.
3. Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. 2013. The emerging field of signal processing on graphs. IEEE Signal Process Mag. 30(3):83–98.
4. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. 2017. Neural message passing for quantum chemistry. Proc Int Conf Mach Learn. 70:1263–1272.
5. Xu K, Hu W, Leskovec J, Jegelka S. 2019. How powerful are graph neural networks? Proc Int Conf Learn Represent.
6. Kipf TN, Welling M. 2017. Semi-supervised classification with graph convolutional networks. Proc Int Conf Learn Represent.
7. Hamilton WL, Ying R, Leskovec J. 2017. Inductive representation learning on large graphs. Proc Adv Neural Inf Process Syst. 30:1024–1034.
8. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. 2018. Graph attention networks. Proc Int Conf Learn Represent.
9. Morris C, Ritzert M, Fey M, et al. 2019. Weisfeiler and Leman go neural: higher-order graph neural networks. Proc AAAI Conf Artif Intell. 33(1):4602–4609.
10. Klicpera J, Bojchevski A, Günnemann S. 2019. Predict then propagate: graph neural networks meet personalized PageRank. Proc Int Conf Learn Represent.
11. Oono K, Suzuki T. 2020. Graph neural networks exponentially lose expressive power for node classification. Proc Int Conf Learn Represent.
12. Ruiz L, Gama F, Ribeiro A. 2020. Graphon neural networks and the transferability of graph neural networks. Proc Adv Neural Inf Process Syst. 33:1702–1712.