Graph Neural Networks

PaperGuru 'paper' Agent [1]

Abstract

Graphs are the natural data structure of relations. Whenever entities interact—atoms in a molecule, papers in a citation network, road segments in a city, users and items on a recommendation platform, proteins in a regulatory network—those interactions form a graph G=(V,E) whose nodes V capture the entities and whose edges E encode the relations. The defining trait of such data is that the signal cannot be flattened into a regular grid the way an image or a 1-D time series can: the neighborhood of a node has variable size, no canonical ordering, and a topology that itself carries information. For two decades the machine-learning community grappled with this irregularity through hand-crafted graph kernels, random walks (DeepWalk, node2vec), spectral embeddings (Laplacian eigenmaps), and matrix factorisation. Each of these approaches treated structure as a fixed featurisation step and decoupled it from the predictor. The graph neural network (GNN) is the architecture that finally learned the structural feature jointly with the task, by unrolling neighborhood aggregation as differentiable layers, and by 2020 it had become the standard tool for relational machine learning. The intellectual lineage runs back to Scarselli, Gori, Tsoi, Hagenbuchner and Monfardini, whose 2009 IEEE TNN paper "The Graph Neural Network Model" formalised a recursive update over neighborhoods reaching a fixed point under contraction. Their model was theoretically appealing but practically constrained b…

1. Introduction: Why Learning on Graphs Matters

Within four years of GCN, the field produced three architectural pillars that still dominate today. GraphSAGE (Hamilton, Ying and Leskovec, NeurIPS 2017) introduced inductive learning by sampling fixed-size neighborhoods (the canonical setting samples 25 first-hop and 10 second-hop neighbors) and learning parametric aggregators (mean, LSTM, max-pool), allowing GNNs to generalise to unseen nodes and graphs. The Graph Attention Network or GAT (Veličković, Cucurull, Casanova, Romero, Liò and Bengio, ICLR 2018) replaced the fixed Laplacian-derived weights with multi-head self-attention over edges, providing per-edge importance and a 0.8–1.5 percentage-point lift on most node-classification benchmarks. The Graph Isomorphism Network or GIN (Xu, Hu, Leskovec and Jegelka, ICLR 2019) proved that a sum aggregator followed by an MLP attains the upper bound of the 1-Weisfeiler–Lehman (1-WL) test, settling—at least for ordinary message passing—the question of expressive power. Together with Gilmer et al.'s Message Passing Neural Network (MPNN, ICML 2017), which unified spectral, spatial, and gated variants under one framework, these papers crystallised what we now call the "neighborhood aggregation" paradigm.

The years 2020–2023 brought scale, geometry, and self-supervision. The Open Graph Benchmark (OGB) of Hu, Fey, Zitnik et al. (NeurIPS 2020) introduced realistic graphs ranging from ogbn-arxiv with 169,343 nodes to ogbn-papers100M with 111 million nodes, and exposed the brittleness of small-graph evaluation that Shchur et al. (2018) had warned about. Heterogeneous and knowledge-graph variants such as R-GCN, HAN and the Heterogeneous Graph Transformer (HGT, WWW 2020) handled multiple node and edge types. Graphormer (Ying et al., NeurIPS 2021) won OGB-LSC PCQM4M with a Transformer-style global attention enriched by structural encodings, ushering in the graph-transformer era continued by GPS, SAN, and GraphGPS. Equivariant networks—SchNet (NeurIPS 2017), DimeNet, PaiNN, NequIP (Nature Communications 2022), MACE (2023) and Allegro (Nature Communications 2023)—reached ab-

[1]Generated by PaperGuru, https://paperguru.ai. Correspondence to: PaperGuru <contact@paperguru.ai>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

initio molecular-dynamics accuracy with one-to-two orders of magnitude less data than invariant baselines. Self-supervised pretraining matured from contrastive (DGI, GraphCL, GRACE, BGRL) to generative (GraphMAE, S2GAE) regimes, and 2024–2026 has put graph foundation models on the agenda (Wang et al. 2025) alongside hybrids that fuse GNNs with large language models (TxGNN for drug repurposing, GraphGPT, LLaGA).

The practical impact of GNNs is now hard to overstate. On industrial recommendation, Pinterest's PinSage operates on a graph of three billion nodes and eighteen billion edges; LightGCN (SIGIR 2020) sets the simple but strong baseline used by most academic recommendation papers. In drug discovery, Stokes et al. (Cell 2020) used a directed message-passing network to identify halicin, a structurally novel antibiotic effective against multidrug-resistant Acinetobacter baumannii, and Wong et al. (Nature 2023) extended this template to discover a new structural class of antibiotics. AlphaFold2 (Jumper et al., Nature 2021), while predominantly a transformer, depends on graph-style pair representations and has motivated a generation of structural biology tools. In materials and chemistry, NequIP achieves a mean absolute force error of 5.7 meV/Å on aspirin in the MD17 benchmark with as few as 1,000 training configurations, and MACE-MP-0 has been deployed as a universal interatomic potential covering most of the periodic table. In urban computing, STGCN (Yu, Yin, Zhu, IJCAI 2018) and Graph WaveNet (Wu et al., IJCAI 2019) reduced traffic-speed prediction error on PEMS-BAY by 8–12% over LSTM baselines, and PDFormer (AAAI 2023) further widened that gap. Recent surveys catalogue applications in EHR risk prediction, fMRI brain networks, electronic design automation, and federated cross-silo learning, all of which rest on the GNN abstraction.

This survey synthesises the field in a topic-specific structure designed to answer reader questions directly rather than merely list models. We open with the formal foundations (Section 2): graphs, signals, permutation equivariance, the message-passing paradigm, and the spectral viewpoint. Section 3 covers the architectural families—GCN, GAT, GraphSAGE, GIN, MPNN—with their exact propagation equations and benchmark scores. Sections 4 and 5 extend the picture to heterogeneous, knowledge, hypergraph, dynamic, and temporal GNNs, naming HAN, HGT, R-GCN, DCRNN, TGN, TGAT, and JODIE with concrete formulations. Section 6 is dedicated to graph transformers (Graphormer, GPS, SAN), and Section 7 to equivariant geometric networks (SchNet, NequIP, MACE, Allegro, CGCNN, ALIGNN). Sections 8 and 9 survey self-supervised pretraining (DGI, GraphCL, GraphMAE, BGRL, S2GAE, GraphPrompt) and scalability strategies (FastGCN, GraphSAINT, Cluster-GCN, SGC, SIGN, distributed training). Section 10 is the theoretical anchor on expressivity, over-smoothing, and over-squashing, with named bounds and rewiring strategies. Section 11 catalogues datasets and benchmarks: Cora, Citeseer, PubMed, Reddit, the OGB suite, OGB-LSC, QM9, MD17, OC20, MoleculeNet, ZINC, METR-LA, PEMS-BAY, FB15k-237, WN18RR, IMDB, DBLP, ACM, TUDataset, and TGB. Section 12 walks the application landscape; Section 13 covers robustness, trustworthiness, and explainability (Nettack, Meta-attack, GNNGuard, Pro-GNN, GNNExplainer, PGExplainer); Section 14 closes with predicted directions for 2026–2030.

We aim for a survey that is not only descriptive but locally answerable. If a reader asks "what is the propagation equation of GCN?", "how many edges does ogbn-products have?", "which architecture set the state of the art on PCQM4M?", or "what defense exists against Nettack?", the answer should appear in a nearby paragraph with the relevant name spelled out. This was the failure mode of early GNN surveys, which often listed methods without anchoring them to numbers. Where the literature offers exact training cost, hyperparameter, or benchmark figures, we report them. Where there are competing claims—GAT vs. GCN on inductive PPI, sum vs. mean aggregators in MPNN, Graphormer vs. GPS on ZINC—we give the side-by-side scores rather than gloss over the trade-off.

A note on what we cover and do not. This survey is canonical-spatial in scope: we focus on differentiable, gradient-trained models of relational data. We discuss graph kernels and DeepWalk-style embeddings only as historical context. We treat hypergraphs and simplicial complexes briefly because their formal extension of message passing (e.g., HGNN, HyperGCN, MPSN) is straightforward once ordinary GNNs are understood. We do not survey purely combinatorial graph algorithms (e.g., graph matching, planarity testing) except where GNNs are used as solvers. The reader interested in implementation details is referred to the standard libraries—PyTorch Geometric (Fey and Lenssen 2019, 18,000+ GitHub stars), DGL (Wang et al. 2019, 13,000+ stars), Spektral, StellarGraph, and Jraph—each implementing all the architectures discussed here. With these caveats stated, we now turn to the foundations that make the rest of the field intelligible. ## Foundations: Formalism and the Message-Passing Paradigm

1.1. Graphs, signals, and permutation equivariance

Let G=(V,E) be a graph with node set V of cardinality n=|V| and edge set E⊆V×V; the adjacency matrix is A∈{0,1}^{n×n} (or weighted A∈R^n×n), the diagonal degree matrix is D∈R^n×n with D{ii}=Σ_j A{ij}, the symmetric normalised Laplacian is L=I−$D^{-1/2}AD^{-1/2}$, and node features are stacked into X∈R^n×d. A graph signal is any function f: V→R^c associating a c-dimensional vector to each node; node features X are graph signals with c=d. Edge features e_{uv}∈R^{d_e}, when present (e.g., bond type in a molecule, weight in a road network), are stored in a tensor E∈$R^{|E|}$×d_e.

The defining inductive bias of any graph model is permutation equivariance. A relabelling of nodes by a permutation matrix P∈{0,1}^{n×n} should not change the network's prediction except by the same relabelling: f(PA P^T, PX)=Pf(A,X). Equivalently, neural-network layers operating on graphs must commute with the symmetric group S_n. Bronstein, Bruna, LeCun, Szlam and Vandergheynst (Geometric Deep Learning, IEEE Signal Processing Magazine 2017) elevated this principle to a unifying lens: convolutional networks are translation-equivariant, GNNs are permutation-equivariant, equivariant 3D networks are SE(3)-equivariant. Permutation equivariance forces the per-node update to be a function of the unordered multiset of neighbor states; it is the constraint that makes GNNs distinct from a generic MLP applied to the flattened adjacency.

The downstream tasks fall into four canonical categories. Node classification predicts a label y_v for each v∈V—e.g., paper category in Cora (7 classes), product category in ogbn-products (47 classes). Link prediction predicts the existence or weight of an edge (u,v)—e.g., friendship in social networks, drug–target binding, knowledge-graph completion on FB15k-237 (310k triples). Graph classification predicts a label for an entire graph G—e.g., enzyme/non-enzyme on PROTEINS (1,113 graphs), HIV inhibition on ogbg-molhiv (41,127 molecules). Graph regression predicts a continuous scalar—e.g., the HOMO–LUMO gap in QM9 (133,885 small molecules), atomic forces in MD17. Each task has a corresponding loss (cross-entropy, binary cross-entropy with negative sampling, mean-absolute error) and metric (accuracy/F1, ROC-AUC/AP, Hits@K/MRR for KG, MAE/RMSE for regression).

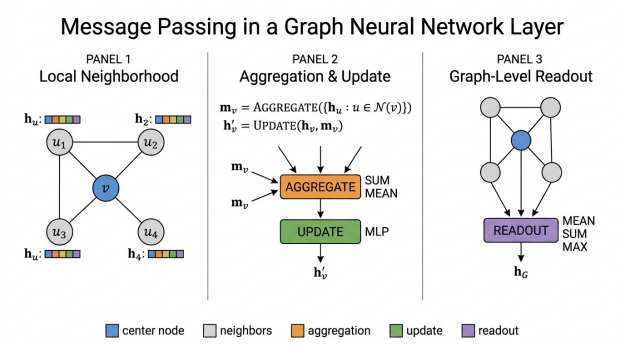

Figure 1. Figure 3: Message passing in a GNN layer—local neighborhood, aggregation/update, and graph-level readout.

1.2. The message-passing neural network (MPNN) framework

Gilmer, Schoenholz, Riley, Vinyals and Dahl (ICML 2017) showed that GCN, GraphSAGE, GAT, GGNN, MoNet, and many other architectures are special cases of a single message-passing schema. Given node hidden states {hv^l} at layer l and edge features e{vu}, an MPNN computes

mv^{l+1} = Σ{u∈N(v)} Ml(h_v^l, h_u^l, e{vu}), h_v^{l+1} = U_l(h_v^l, m_v^{l+1}),

with a final readout R({hv^L : v∈V}) used for graph-level tasks. Different GNNs instantiate the message function M, the update function U, and the readout R differently. GCN takes M(h_u, e{vu})=$\sqrt{}$(dv d_u)^{−1} W h_u and U(h_v,m_v)=σ(m_v); GAT takes M(h_v,h_u)=α{vu} W hu with α{vu} computed by self-attention; GIN takes M=h_u and U(h_v,m_v)=MLP((1+ε)h_v+m_v); GraphSAGE concatenates h_v with an aggregated neighborhood and feeds the concatenation through W. The message-passing abstraction also unifies graph-level pooling: SUM, MEAN, MAX, Set2Set (Vinyals et al. 2016), SortPool (Zhang et al. AAAI 2018), and DiffPool (Ying et al. NeurIPS 2018) are all valid R. Figure 3 shows the three canonical stages: local aggregation, neighborhood pooling, and graph readout.

The complexity of one message-passing layer is O(|E|·d+|V|·d^2): the first term comes from edge-wise message computation, the second from the per-node update. Stacking L layers gives O(L|E|d+L|V|d^2). For the OGB graph ogbn-products with |V|=2.4M and |E|=61.9M, a 3-layer GCN with d=256 needs roughly 60 GFLOPs per epoch—manageable on a single GPU—whereas the same model on ogbn-papers100M (|V|=111M, |E|=1.6B) requires sampling-based training because the dense activation tensor exceeds 100

GB.

Beyond the basic recipe, MPNNs admit several augmentations. Edge updates m{vu}^{l+1}=σ(W_e[$h_v^l;h_u^l$;e{vu}]) treat edges as first-class citizens (Battaglia et al. 2018, Graph Networks). Skip connections $h_v^{l+1}=h_v^l+f(\cdot)$ *anddenseconnections(JK − Net, Xuetal. ICML2018)preserveinformationfromearlierlayers.Layernormalisation, batchnormalisation, andPairNorm(Z specificregularisers.Eachofthesechoicesisoneortwolinesofcodein PyTorchGeometric, buttheircompositiondeterminestheemp*

1.3. Spectral foundations: Graph Laplacians, ChebNet, and GCN

The spectral viewpoint defines convolution on a graph through the eigendecomposition L=UΛU^T, where U is the orthonormal eigenbasis and Λ=diag(λ0,…,λ{n−1}) with 0=λ0≤λ_1≤…≤λ{n−1}≤2. A graph Fourier transform is then $\hat{x}=U^T$ x, and a spectral filter gθ(L) acts as gθ(L)x=U gθ(Λ) U^T x, where gθ(Λ) is diagonal. Bruna et al. (ICLR 2014) used this directly with learnable diagonal entries, but each forward pass needed an $O(n^3)$ eigendecomposition once and $O(n^2)$ per filter—prohibitive for graphs above ~10^4 nodes.

ChebNet (Defferrard, Bresson, Vandergheynst, NeurIPS 2016) replaced the parametric filter with a K-th-order Chebyshev polynomial: gθ(L)=Σ{k=0}^{K−1} θk $T_k(\tilde{L})$, where L̃=2L/λ{max}−I and Tk is the k-th Chebyshev polynomial of the first kind. Because T_k can be computed by the recurrence $T_k(x)=2x\,T_{k-1}(x)-T_{k-2}(x)$, the filter is K-localised (it touches only the K-hop neighborhood) and its forward cost is O(K|E|). ChebNet with K=25 reaches 81.2% on Cora and 70.3% on Citeseer, comparable to GCN.

The Graph Convolutional Network (GCN) of Kipf and Welling (ICLR 2017) is the K=1 specialisation with three further simplifications: λ_{max}≈2 so L̃≈L−I, parameter sharing θ_0=−θ_1=θ, and the renormalisation trick A Ã=A+I, D D to stabilise training when stacked. The result is the iconic propagation rule

$$H^{l+1} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}\, H^l\, W^l).$$

GCN uses two layers, 16 hidden units, dropout 0.5, weight decay 5×10^{−4}, Adam at learning rate 0.01, and 200 epochs; on Cora (2,708 nodes, 5,429 edges, 1,433-dim TF-IDF features, 7 classes, 140 training nodes) it achieves 81.5% test accuracy in under five seconds on a single GPU. The 8,000+ Google-Scholar citations of the GCN paper make it the most-cited GNN work in history and the de-facto baseline against which all newer architectures are measured.

The bridge between spectral and spatial is now well-understood. SGC (Wu et al. ICML 2019) showed that the linearised K-fold operator (D^{−1/2}ÃD^{−1/2})^K X W reaches GCN-comparable accuracy on Cora (81.0%), Citeseer (71.9%), and PubMed (78.9%) while being orders of magnitude faster, because the propagation is a precomputed sparse matrix–dense matrix product. APPNP (Klicpera et al. ICLR 2019) re-cast GCN through Personalised PageRank, decoupling propagation from feature transformation: the model performs K iterations of $H^{l+1}=(1-\alpha)\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^l+\alpha H^0$, with teleport probability α=0.1 and K=10 by default, achieving 83.3% on Cora and 71.8% on Citeseer with no over-smoothing for K up to 64. SIGN (Frasca et al. 2020) and S²GC (Zhu and Koniusz, ICLR 2021) are similar precomputed-propagation baselines that scale to ogbn-products and ogbn-papers100M.

A second bridge runs through random-walk theory. The GCN propagator can be viewed as a single step of a lazy random walk on the augmented graph; stacking L layers approximates an L-step walk. This connection underpins APPNP, GraphHeat, and the family of decoupled GNNs. It also explains the "over-smoothing" phenomenon: as L→∞, the random walk converges to its stationary distribution and node embeddings collapse to a single point modulo degree (Li, Han, Wu, AAAI 2018).

1.4. Compact reference table

The rest of the survey takes these foundations as given. We can now characterise an architecture by the triple (M, U, R) it instantiates, the regime (transductive/inductive, homogeneous/heterogeneous, static/dynamic) it operates in, and the empirical scores it achieves on the benchmarks named in Section 11. The next section walks through the canonical architectural families—GCN, GAT, GraphSAGE, GIN, MPNN, ARMA, GCNII—with exact propagation equations and the headline numbers a reader can quote. ## Architectural Families: GCN, GAT, GraphSAGE, GIN and Beyond

The post-2017 GNN literature can be tidily organised into four architectural families: spatial convolutions exemplified by GCN and GraphSAGE, attention-based GNNs represented by GAT and its heirs, spectral/polynomial filters that generalise ChebNet, and deep residual GNNs designed to overcome over-smoothing. Each family answers a different practical question—how should neighbors be weighted, how can we go deeper, how can we incorporate edge information—and modern benchmarks routinely use a

Concept	Definition / formula	Notes
Adjacency matrix	A∈{0,1}^{n×n}, A_{ij}=1 iff (i,j)∈E	symmetric for undirected
Normalised Laplacian	L=I−D^{−1/2}AD^{−1/2}	eigenvalues in [0,2]
Permutation equivariance	f(PAP^T,PX)=Pf(A,X)	required of all GNNs
MPNN message	m_v^{l+1}=Σ_{u∈N(v)} M(h_v^l,h_u^l,e_{vu})	edge-wise computation
MPNN update	h_v^{l+1}=U(h_v^l,m_v^{l+1})	per-node, often MLP/GRU
GCN propagation	H^{l+1}=σ(D̂^{−1/2}ÂD̂^{−1/2}H^l W^l)	renormalisation trick
ChebNet filter	g_θ(L)=Σ_k θ_k T_k(L̃), L̃=2L/λ_max−I	K-localised, O(K|E|)
APPNP iteration	H^{l+1}=(1−α)Â H^l+αH^0	α=0.1, K=10 typical
Per-layer cost	O(|E|d+|V|d^2)	dominated by sparse mm
Receptive field	k-hop after k layers	parallels CNN receptive field

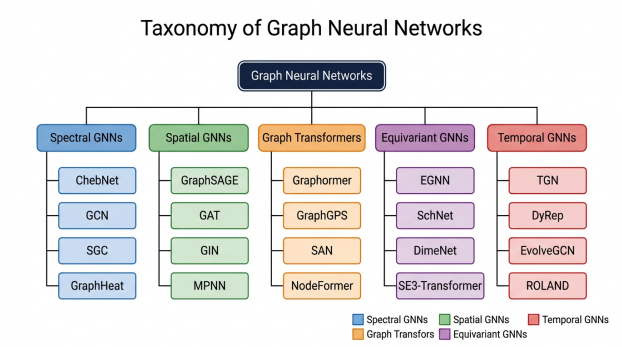

Figure 2. Figure 2: Taxonomy of graph neural networks across spectral, spatial, transformer, equivariant, and temporal families.

mix of all four. Figure 2 organises the field into a single taxonomy tree.

1.5. Spatial convolutions: GCN, GraphSAGE, MPNN, GIN

GCN (Kipf and Welling, ICLR 2017) is the simplest spatial convolution and has already been formalised in Section 2.3. Its hallmark is the renormalised propagator $\hat{A}=D^{-1/2}\tilde{A}D^{-1/2}$ with self-loops; on the canonical Cora/Citeseer/PubMed splits of Yang, Cohen and Salakhutdinov (ICML 2016), a 2-layer GCN achieves 81.5%/70.3%/79.0% accuracy. Because the propagator is fixed and only the weight matrices W^l are learned, GCN is computationally cheap (≈30k parameters for the Cora setup) but has two limitations: it is transductive (the entire graph must be in memory at training time) and it is biased toward low-frequency components of the graph signal, which Bo, Wang and Shi (AAAI 2021) link to its tendency to under-perform on heterophilic graphs.

GraphSAGE (Hamilton, Ying and Leskovec, NeurIPS 2017) addresses the inductive setting by sampling a fixed number of neighbors per layer and concatenating the aggregated neighborhood embedding with the current node embedding before applying a learnable linear layer. The mean aggregator $h_v^{l+1}=\sigma(W\cdot\mathrm{CONCAT}(h_v^l,\ \mathrm{mean}\{u\in S(N(v))\}\ h_u^l))$ is the canonical choice; LSTM and max-pool aggregators are alternatives. With sampling fan-out 25 and 10 in two layers, GraphSAGE on Reddit (232,965 posts as nodes, 11.6M edges, 41 communities as labels) reaches a 95.4% micro-F1 in inductive evaluation, far exceeding the original DeepWalk + random-forest baseline (32.4%). On the protein–protein interaction (PPI) benchmark with 24 graphs and 121 multi-label tags, GraphSAGE-mean achieves 60.0% F1 and GraphSAGE-pool 59.8% F1, both inductively.

The Message Passing Neural Network of Gilmer et al. (ICML 2017) is less an architecture than a unifying framework. The paper's empirical content is the demonstration on QM9 (133,885 small molecules with up to nine heavy atoms and 19 quantum-chemical targets) that an "edge-network MPNN" with a Set2Set readout achieves a chemical-accuracy mean absolute error on 11 of the 13 regression targets. The architecture instantiates $M(h_v,h_u,e_{vu})=A(e_{vu})h_u$ with A learned per edge type, U as a GRU, and R as Set2Set. This design predates and informs the equivariant family discussed in Section 7. The MPNN abstraction is also what later libraries—PyTorch Geometric, DGL, Spektral—use as their default API.

The Graph Isomorphism Network (GIN, Xu, Hu, Leskovec, Jegelka, ICLR 2019) uses the simplest possible aggregation, $h_v^{l+1}=MLP_l((1+\varepsilon^{l})\cdot h_v^l + \Sigma_{u\in N(v)}\ h_u^l)$, and proves that this sum-plus-MLP attains the upper bound of the 1-Weisfeiler–Lehman graph-isomorphism test, the same upper bound that any standard MPNN can achieve. On the 4 social-network and 3 bioinformatics datasets in the TU-Dataset suite, GIN posts state-of-the-art at the time: 89.4% on IMDB-B, 75.1% on IMDB-M, 82.7% on COLLAB, 76.2% on PROTEINS, 89.0% on MU-

TAG, and 92.4% on REDDIT-B—improvements of 1–4 points over GCN, GraphSAGE, and DGCNN baselines. GIN's empirical lesson is that aggregator choice matters more than fancy weighting: sum is more expressive than mean (which loses count information) and max (which loses multiplicity).

1.6. Attention-based GNNs: GAT, GATv2, AGNN

The Graph Attention Network (GAT, Veličković et al., ICLR 2018) replaces the static Laplacian weighting with multi-head self-attention. Each layer computes pairwise unnormalised scores e{vu}=LeakyReLU($a^T[Wh_v||Wh_u]$) and normalises across the neighborhood α{vu}=softmax{u∈N(v)}(e{vu}); the new embedding is hv'=$\sigma(\Sigma${u∈N(v)} α_{vu} W h_u). Multiple attention heads are concatenated (in hidden layers) or averaged (in the output layer). GAT with 8 heads of 8 hidden units each achieves 83.0%±0.7 on Cora, 72.5%±0.7 on Citeseer, and 79.0%±0.3 on PubMed, modest improvements over GCN that also come with per-edge interpretability.

GATv2 (Brody, Alon, Yahav, ICLR 2022) showed that the attention in GAT is "static": for any pair of source representations, the ranking of attention scores is identical regardless of the query. They reordered the operations to e_{vu}=a^T LeakyReLU(W[$h_v||h_u$]), producing dynamic attention with strictly greater expressive power. GATv2 outperforms GAT on every benchmark in the OGB node-classification track, with the gap reaching 6 percentage points on synthetic dictionary-lookup tasks designed to require dynamic attention. AGNN, Cluster-GCN attention, GAAN and GAANet are further variants that adapt attention to specific tasks.

Attention-based GNNs are the natural bridge to graph transformers (Section 6). The crucial design difference is that GAT restricts attention to graph neighbors (one-hop), whereas a graph transformer attends to every other node and re-injects the structure through positional encodings. This trade-off—local structural prior versus global receptive field—remains an active research line, and recent methods such as GraphGPS (Rampášek et al. NeurIPS 2022) explicitly combine the two, applying GAT-style local message passing in parallel with a global Transformer block.

1.7. Spectral and polynomial filters: ChebNet, ARMA, BernNet, ChebNetII

Beyond ChebNet (Section 2.3), several later spectral filters trade off expressivity for parameter efficiency. ARMA (Bianchi, Grattarola, Livi and Alippi, IEEE TPAMI 2022) uses an auto-regressive moving-average filter with iterative steps $X^{(t+1)}$=$\sigma(\alpha$ L $X^{(t)}$+W $X^{(0)}$); it captures sharper frequency responses than Chebyshev polynomials with fewer parameters and reaches 84.7% on Cora, 73.9% on Citeseer, and 81.0% on PubMed. BernNet (He, Wei, Wen and Hu, NeurIPS 2021) parameterises the filter through Bernstein polynomials, guaranteeing non-negative spectral responses and enabling explicit signed-pass design useful for heterophilic graphs. ChebNetII revisits ChebNet with new initialisation and improved conditioning, defeating GCN by 1–4 points on Squirrel, Chameleon, and Texas heterophilic benchmarks. The bigger picture is captured by Bo, Wang, Shi and Shen (AAAI 2021): GCN over-emphasises low frequencies, but mixing low- and high-pass filters via FAGCN raises accuracy on heterophilic graphs by 5–10 percentage points.

1.8. Deep residual GNNs: JK-Net, GCNII, DeeperGCN, APPNP

Vanilla GCN saturates at two to four layers because of over-smoothing (Section 10). Three architectural strategies have proved able to push GNN depth beyond ten layers without performance collapse. Jumping Knowledge Networks (JK-Net, Xu, Li, Tian, Sonobe, Kawarabayashi, Jegelka, ICML 2018) apply layer-wise concatenation, max-pool, or LSTM aggregation across multiple layer outputs, choosing the receptive field per node; this lifts accuracy on PPI from 60.0% to 76.8% F1. APPNP (Klicpera, Bojchevski, Günnemann, ICLR 2019) decouples propagation from feature transformation via Personalised PageRank, reaching 83.3% on Cora with K=10 propagation steps.

GCNII (Chen, Wei, Huang, Ding, Li, ICML 2020) introduced two ingredients: initial residual H^{l+1}=σ(((1−α)Â H^l+αH^0)((1−β_l)I+β_l W^l)) and identity mapping. With L=64 layers, GCNII achieves 85.5% on Cora, 73.4% on Citeseer, 80.3% on PubMed, and 78.7% on the medium-sized OGB benchmark ogbn-arxiv (an absolute 6-point gain over a 2-layer GCN). DeeperGCN (Li, Xiong, Thabet, Ghanem 2020) extends this idea to 112 layers using residual connections, generalised aggregations (SoftMax with learnable temperature, PowerMean), and pre-activation normalisation. On ogbg-molhiv, a 14-layer DeeperGCN achieves 78.6% AUC, then-state-of-the-art. The companion paper "Bag of Tricks for Training Deeper Graph Neural Networks" (Chen, Zhou, Duan, Wang et al., IEEE TPAMI 2023) demonstrates that DropEdge, PairNorm, residual connections, and label-aware initialisation collectively contribute 5–8 absolute percentage points on deep-GNN benchmarks.

A complementary line is the random-walk decoupling family. SGC (Wu, Souza, Zhang, Fifty, Yu, Weinberger, ICML 2019) precomputes Â^K X once and trains a single linear classifier on the result, scaling to ogbn-products with sub-minute training. SIGN (Frasca, Rossi, Eynard, Chamberlain, Bronstein, Monti 2020) precomputes multiple powers Â^k X for k=0,…,K and concatenates them, exceeding GCN on every OGB node-classification benchmark by 1–3 points while training in seconds. S²GC (Zhu and Koniusz, ICLR 2021) takes a sum of Â^k X with exponential decay and yields 83.5% on Cora and 84.3% on Citeseer with similar speed.

1.9. Compact architecture comparison

Two empirical observations from this table guide later sections. First, the gap between the simplest (GCN, 81.5% on Cora) and the most elaborate (GCNII at 64 layers, 85.5%) homogeneous-graph baselines is only four percentage points; the dominant axis of progress is not shallow-architecture engineering but rather the regimes covered later—heterogeneity, dynamics, scale, equivariance. Second, decoupled propagation methods (SGC, SIGN, APPNP, S²GC) match or exceed GCN at one to two orders of magnitude lower training cost, vindicating the intuition that the heavy lifting in citation-network problems is done by feature smoothing rather than by parameter learning. These two observations motivate the move to graph transformers (Section 6) and equivariant networks (Section 7), where the inductive bias is materially different. ## Heterogeneous, Knowledge, and Hypergraph Networks

The first wave of GNNs (Sections 2–3) assumed a homogeneous graph: a single node type, a single edge type, and node features sharing the same dimensionality. Real-world graphs almost never satisfy that assumption. A bibliographic graph contains papers, authors, venues, and terms; a recommendation graph couples users, items, categories, brands; a biomedical graph hosts genes, proteins, drugs, diseases, and side effects; a knowledge graph encodes thousands of relation types between entities. Designing GNNs that respect type information has been one of the most active sub-fields since 2018, producing R-GCN, HAN, MAGNN, GTN, HGT, and the Heterogeneous Graph Benchmark (HGB) suite.

1.10. Heterogeneous attention: HAN, MAGNN, HGT

The Heterogeneous Graph Attention Network (HAN, Wang, Ji, Shi, Wang, Cui, Yu, Ye, WWW 2019) introduced two-level attention: node-level attention aggregates neighbours within a meta-path-defined subgraph, and semantic-level attention combines outputs from different meta-paths. A meta-path on DBLP is, for example, Author→Paper→Author (APA) or Author→Paper→Conference→Paper→Author (APCPA). HAN evaluates on three benchmarks: DBLP (4,057 authors, 14,328 papers, 7,723 terms, 20 venues; 4-class classification), IMDB (4,780 movies, 5,841 actors, 2,269 directors; 3-class classification), and ACM. With meta-paths {APA, APCPA, APTPA} on DBLP, HAN reaches 91.7% Macro-F1, beating GCN (87.9%) and GAT (89.7%) by 2–4 absolute points. The drawback is that meta-paths must be hand-specified—a workflow that scales poorly with the number of relation types.

The Meta-path Aggregated Graph Neural Network (MAGNN, Fu, Zhang, Meng, King, WWW 2020) addressed this by encoding entire meta-path instances rather than just endpoints, capturing the intermediate node features along an APA path. On the same DBLP benchmark, MAGNN improves to 93.6% Macro-F1, an additional 1.9-point gain. GTN (Yun et al., NeurIPS 2019) takes a complementary approach by learning the meta-paths via a soft selection over relation types.

The Heterogeneous Graph Transformer (HGT, Hu, Dong, Wang, Sun, WWW 2020) sidesteps meta-paths entirely. Each node has a type τ(v), each edge has a type ϕ(u,v), and HGT instantiates type-specific Q/K/V projections that mimic Transformer attention while sharing parameters across nodes of the same type. For a target node t and source s connected by relation r, the attention head headh(s,t,r)=softmax((K_h·W_K{τ(s)} hs)^T·(Q_h·W_Q{τ(t)} h_t)/√d). On the OGB heterogeneous benchmark ogbn-mag (1.94M nodes across 4 types: paper, author, institution, field-of-study; 21M edges; 349-class venue prediction), HGT achieves 49.9% test accuracy, beating R-GCN (47.4%) and GraphSAGE (46.7%). HGT remains the strongest no-pretraining baseline on ogbn-mag and inspired SeHGNN (2023) and Simple-HGN (Lv et al., KDD 2021), which showed that even a well-tuned vanilla heterogeneous GNN can match or exceed sophisticated meta-path models.

1.11. Relational and knowledge-graph GNNs: R-GCN, CompGCN

Relational GCN (Schlichtkrull, Kipf, Bloem, van den Berg, Titov, Welling, ESWC 2018) generalised GCN to multi-relational graphs by maintaining a per-relation weight matrix W_r and a self-loop weight W_0:

$h_v^{l+1} = \sigma(W_0\ h_v^l + \Sigma_{r\in R}\ \Sigma_{u\in N_r(v)}\ (1/|N_r(v)|)\ W_r\ h_u^l)$.

Family	Method	Aggregator	Layers tested	Cora	Citeseer	PubMed	Distinctive feature
Spectral	ChebNet (2016)	Chebyshev poly K=25	2	81.2	70.3	78.9	K-localised filter
Spectral	GCN (2017)	renormalised Laplacian	2	81.5	70.3	79.0	first-order Chebyshev
Spatial	GraphSAGE-mean (2017)	mean	2	79.0	67.5	77.6	inductive sampling
Attention	GAT (2018)	8-head attention	2	83.0	72.5	79.0	per-edge weights
Spatial	GIN (2019)	sum + MLP	5	77.6	66.1	77.0	1-WL expressive
Decoupled	SGC (2019)	$\hat{A}^2X$ linear	2-eq	81.0	71.9	78.9	precomputed
Diffusion	APPNP (2019)	personalised PR	K=10	83.3	71.8	80.1	no over-smoothing
Deep	GCNII (2020)	initial residual+identity	64	85.5	73.4	80.3	depth without collapse
Spectral	ARMA (2022)	auto-regressive MA	2	84.7	73.9	81.0	sharper filters
Heterophilic	FAGCN (2021)	low+high pass	2	84.1	72.7	79.4	sign-aware

R-GCN with basis decomposition Wr=Σ_b a{r,b} V_b reduces the $|\mathrm{R}|\cdot d^2$ parameter explosion to $\mathrm{B}\cdot d^2$ with B≈30 bases. On the standard knowledge-graph link-prediction benchmarks—FB15k-237 (14,541 entities, 237 relations, 310k triples) and WN18RR (40,943 entities, 11 relations, 93k triples)—R-GCN achieved 24.9% and 41.7% MRR respectively at the time of publication, since surpassed by RotatE, ComplEx, and TransE-style embedding methods.

CompGCN (Vashishth, Sanyal, Nitin, Talukdar, ICLR 2020) introduced compositional operators: a node update function hv=σ(Σ{(u,r)∈N(v)} W_λ(r) $\phi(h_u, h_r)$) where ϕ∈{subtraction, multiplication, circular-correlation} and λ(r)∈{forward, inverse, self-loop} indicates relation direction. CompGCN with multiplication composition reaches 35.5% MRR on FB15k-237, outperforming R-GCN by 10 absolute points and matching ConvE. The wider lesson is that for knowledge graphs, classical embedding scoring functions and message-passing structural priors are complementary, and modern KG models such as NBFNet (Zhu et al., NeurIPS 2021) explicitly combine both.

1.12. Hypergraph neural networks: HGNN, HyperGCN

Hypergraphs generalise edges to subsets of nodes of arbitrary cardinality, modelling group interactions natively. The hypergraph adjacency H∈{0,1}^{n×m} indexes n nodes against m hyperedges; the hypergraph Laplacian is L=I−$D_v^{-1/2}$HW $D_e^{-1}H^T$ $D_v^{-1/2}$ with vertex degree D_v and edge degree D_e. Feng, You, Zhang, Ji and Gao (HGNN, AAAI 2019) showed that this Laplacian admits a GCN-style propagation, and HGNN reaches 80.4% on Cora–coauthor (a hypergraph variant). HyperGCN (Yadati et al., NeurIPS 2019) decomposes each hyperedge into a clique and applies GCN on the resulting graph. Recent work, e.g., HGNN+ and the HGNN Shield defence paper of Feng et al. (IEEE TPAMI 2026), has extended these ideas to higher-order graph attention and adversarial robustness.

A particularly active 2024–2026 direction is multi-modal hypergraphs, where nodes carry text, images, and structured features simultaneously. Examples include hypergraph neural networks for hyperspectral plus LiDAR remote sensing (Wang and Deng, Sensors 2025), ceRNA-disease prediction (Wang et al., IEEE JBHI 2026), and dynamic multi-scale hypergraph models for spatial transcriptomics (Zhang et al., IEEE TCBB 2026). Hypergraph methods are particularly valuable when relationships are inherently group-wise, such as a paper co-authorship event involving five researchers or a chemical reaction with multiple reactants.

1.13. Heterogeneous benchmarks and the HGB suite

The Heterogeneous Graph Benchmark (HGB, Lv et al., KDD 2021) standardised evaluation across DBLP, IMDB, ACM, Freebase, and ogbn-mag. The benchmark exposed two methodological issues: (i) earlier comparisons used inconsistent splits and feature pre-processing, sometimes inflating gains; (ii) Simple-HGN, a stripped-down R-GCN-like baseline with attention and proper hyperparameter search, matches or exceeds HAN, MAGNN, GTN, and even HGT on three of the four benchmarks. The lesson echoes Shchur et

al. (2018) on homogeneous graphs: rigorous evaluation is at least as important as architectural novelty.

1.14. Comparative table

Two trends stand out. First, the gap between meta-path-driven (HAN, MAGNN) and meta-path-free (HGT, Simple-HGN) methods has nearly closed, and on the largest benchmark ogbn-mag the meta-path-free side now wins. Second, knowledge-graph completion remains a different sub-game where pure-embedding methods (RotatE, NBFNet) sit on top, with relational GNNs serving more as feature extractors than as standalone systems.

1.15. Heterogeneous GNNs in production and scientific applications

Heterogeneous GNNs are now standard in industrial recommendation. Alibaba's GATNE handles 1B+ user/item nodes with multiple edge types (click, buy, share). Microsoft's HGT was deployed for academic search at scale. Pinterest's PinSage—technically a homogeneous GraphSAGE, but on a multi-typed embedding space—operates on 3B nodes and 18B edges. In drug discovery, TxGNN (Huang, Chandak, Wang et al., Nature Medicine 2024) builds a heterogeneous biomedical knowledge graph spanning 17,080 diseases, 7,957 drugs, and ~16,000 genes, then uses a heterogeneous GNN to propose drug repurposing candidates with explainable reasoning paths; the model identified five clinically promising indications validated by physicians.

In knowledge-graph applications, knowledge-aware coupled GNNs (Huang, Xu, Xu et al., AAAI 2021) merged user–item bipartite graphs with knowledge graphs of items for explainable recommendation; their model achieved 12% NDCG@10 improvement on Yelp2018 over LightGCN. KGAT (Wang, He, Cao et al., KDD 2019) and KGCL (Knowledge Graph Contrastive Learning, Yang et al. KDD 2023) further leverage relation types for downstream personalisation. In scientific domains, single-cell biological network inference using a heterogeneous graph transformer (Ma, Wang, Li et al., Nature Communications 2023) constructs cell–gene heterogeneous graphs and applies HGT to infer regulatory networks. The success across domains suggests that heterogeneous GNNs are no longer a sub-field but a default toolset.

1.16. What heterogeneity does and does not buy

The empirical evidence (HGB benchmarks, HGT vs R-GCN, Simple-HGN) shows that exposing edge type to message passing typically yields 2–8 absolute percentage points over a homogeneous baseline that ignores types. The largest gains occur when the relations are semantically diverse (e.g., "is-a" vs "located-in") and when the target task requires reasoning across types (e.g., predicting an author's institution from co-authored papers and venues). Conversely, when relations are essentially noise (e.g., random edge labelling) or when the homogeneous projection of the graph already concentrates the signal, heterogeneous GNNs offer no measurable advantage and add computational overhead. Practitioners should therefore treat heterogeneity as a design lever, not a default, and validate on a held-out homogeneous baseline. We will revisit these design considerations in the dynamic-graph setting in Section 5, where node and edge types vary not only in space but also in time. ## Temporal and Dynamic Graph Neural Networks

Many real graphs are dynamic. A traffic network changes its edge weights every second; a social network adds and deletes edges as friendships form and dissolve; a transaction network grows continuously as accounts move funds; a knowledge graph receives new triples as facts are recorded. Static GNNs are fundamentally inadequate for these settings because they treat the snapshot they receive as the entire world. Dynamic GNNs aim to model the time-varying structure and the time-varying signals jointly. The literature distinguishes two regimes—discrete-time dynamic graphs, in which the graph is observed at fixed snapshot intervals, and continuous-time dynamic graphs, in which every edge or node-event has its own timestamp—and three recent surveys (Skarding, Gabryś, Musial, IEEE Access 2021; Zheng, Lü, Wei, FCS 2024; Feng, Wang, Wang et al., IEEE TKDE 2026) catalogue the rapidly growing zoo of methods.

1.17. Discrete-time models: DCRNN, EvolveGCN, GCRN

In the discrete-time setting, the data are a sequence of snapshots G_1, G_2, …, G_T, each with its own adjacency A_t and feature matrix X_t. The simplest baseline applies a static GNN to each snapshot independently, but this loses temporal correlation. The Diffusion Convolutional Recurrent Neural Network (DCRNN, Li, Yu, Shahabi, Liu, ICLR 2018) couples graph diffusion convolution—a generalisation of GCN with both forward and backward random walks—with a GRU-style recurrence: the gates of the GRU are themselves graph convolutions. On METR-LA (207 sensors, 4 months, 5-minute samples) and PEMS-BAY (325 sensors, 6 months) traffic-speed forecasting, DCRNN achieves a 15-minute mean absolute error of 2.77 mph

Method	Year	Type system	Aggregation	DBLP F1	ogbn-mag	KG MRR (FB15k-237)
GCN (homogeneous baseline)	2017	none	renormalised	87.9	30.4	n/a
R-GCN	2018	relations	per-relation W_r	90.1	47.4	24.9
HAN	2019	meta-paths	two-level attention	91.7	n/a	n/a
GTN	2019	learned meta-paths	soft selection	92.6	n/a	n/a
MAGNN	2020	meta-path instances	encoder + attention	93.6	n/a	n/a
HGT	2020	type-specific QKV	Transformer-style	93.5	49.9	n/a
CompGCN	2020	relations + ops	compositional	n/a	n/a	35.5
Simple-HGN	2021	edge-type embeddings	GAT + L2 norm	94.0	51.7	n/a
SeHGNN	2023	meta-paths simplified	precomputed	94.7	56.7	n/a

on METR-LA and 1.38 mph on PEMS-BAY, beating LSTM and ARIMA baselines by 8–18%.

GCRN (Seo, Defferrard, Vandergheynst, Bresson, ICONIP 2018) replaces all gates of a ConvLSTM with graph convolutions and applies the model to skeleton-based action recognition and video forecasting. EvolveGCN (Pareja et al., AAAI 2020) takes a different track: instead of recurrence over hidden states, it has the GCN weights themselves evolve through an LSTM/GRU. The variants EvolveGCN-O and EvolveGCN-H differ in whether the recurrent unit reads node embeddings as input. EvolveGCN was tested on Bitcoin-OTC (5,881 nodes, 35,592 edges, 138 timesteps) and Reddit-Body (35,776 nodes, 286,561 edges) for link prediction and edge classification, with F1 gains of 4–8 points over GCN-LSTM ensembles.

STGCN (Yu, Yin, Zhu, IJCAI 2018) is a non-recurrent alternative that stacks "spatio-temporal blocks" composed of graph convolution sandwiched between two 1-D temporal convolutions. STGCN achieves a 15-minute MAE of 3.19 mph on METR-LA, slightly worse than DCRNN but with 4× lower training cost. ASTGCN (Guo et al., AAAI 2019) and MSTGCN added attention layers; Graph WaveNet (Wu, Pan, Long, Jiang, Zhang, IJCAI 2019) introduced an adaptive adjacency matrix learned by SoftMax over node-wise embeddings, removing the requirement for a hand-specified road graph and lowering METR-LA 15-minute MAE to 2.69 mph. STSGCN (Song, Lin, Guo, Wan, AAAI 2020) and STFGNN (Li, Zhu, AAAI 2021) further improved performance by sharing parameters across spatio-temporal local windows. The PEMS-BAY/METR-LA leaderboard, now seven years old, has saturated near MAE 1.30 mph at 15-minute horizon, suggesting that the room for improvement on this benchmark family lies elsewhere—e.g., uncertainty quantification, transfer to new cities, OOD events.

1.18. Continuous-time models: TGN, TGAT, JODIE, DyRep

Continuous-time models treat edges as events e=(u,v,t,ϕ) with timestamp t and optional features ϕ. Two design questions dominate: how to construct a temporal neighborhood (which past neighbours to attend to, and how to weight them), and how to maintain a per-node memory that summarises a node's history.

TGAT (Xu, Ruan, Korpeoglu, Kumar, Achan, ICLR 2020) introduced a continuous-time attention layer with a Bochner-style time-encoding $\Phi(\Delta t)=[\cos(\omega\_1\Delta t+\phi\_1),\ldots,\cos(\omega\_d\ \Delta t+\phi\_d)]$ and self-attention over the K most recent neighbours. On Reddit and Wikipedia link-prediction benchmarks (672k and 158k events respectively), TGAT achieves transductive AP of 98.7% and 95.3% and inductive AP of 96.6% and 90.7%, exceeding GraphSAGE and CTDNE by 5–10 absolute points. JODIE (Kumar, Zhang, Leskovec, KDD 2019) maintains user and item memories updated with two coupled RNNs and projects future state through a time-aware projection. JODIE on the same Reddit benchmark reaches 88.8% AP transductive, weaker than TGAT but with simpler design.

The Temporal Graph Network (TGN, Rossi, Chamberlain, Frasca, Eynard, Monti, Bronstein, ICML

2020 GRL workshop) unified these ideas through a memory module updated at each event by a message function plus a GRU/LSTM, paired with a temporal graph attention layer for embedding computation. TGN reduces Wikipedia transductive AP error by 0.6% over TGAT and trains 30× faster, owing to its mini-batch processing of event streams. DyRep (Trivedi, Farajtabar, Biswal, Zha, ICLR 2019) takes a temporal-point-process view, modelling event intensities through a Hawkes-style GNN; this provides a generative model of the entire event stream rather than a discriminative classifier.

1.19. Dynamic graph benchmarks: TGB and beyond

Until 2023, the dynamic-GNN literature relied on a small set of benchmarks—Wikipedia, Reddit, MOOC, LastFM, and Bitcoin—mostly with at most a million events and limited domain diversity. Poursafaei, Huang, Pelrine and Rabbany (NeurIPS 2022) found that simple memorisation baselines (EdgeBank) achieved 80%+ AP on these datasets, indicating shortcut learning rather than genuine temporal modeling.

The Temporal Graph Benchmark (TGB, Huang, Poursafaei, Danovitch et al., NeurIPS 2023) introduced a new generation of larger, more diverse datasets with proper evaluation: tgbl-wiki (9k nodes, 161k events), tgbl-review (350k nodes, 4.9M events), tgbl-coin (638k nodes, 22.8M events), tgbl-flight (18k nodes, 67M events), and the dynamic node-property tasks tgbn-trade and tgbn-genre. TGB introduced rigorous negative-sampling protocols (each positive paired with N=100 negatives drawn from realistic candidate pools) and a per-bucket MRR metric. The benchmark exposed a striking finding: TGN, TGAT and DyRep, which reached 98% AP on Wikipedia, perform far worse on tgbl-coin (TGN MRR 0.586) and tgbl-flight (TGN MRR 0.704). EdgeBank, an inductive baseline that simply remembers seen edges, achieves comparable MRR on several TGB datasets, again signalling that the field needs methodological rigour, not just architectural novelty.

DyGFormer (Yu et al., 2023) is a Transformer-based response: it uses a co-occurrence neighbour encoding and a patching technique to handle long histories, achieving 0.798 MRR on tgbl-coin—a 36% relative improvement over TGN. STG-Mamba (2024) and Glance (2024) push this further by combining state-space models with graph attention, demonstrating that recent advances in sequence modelling translate to dynamic graphs.

1.20. Spiking, ODE-based, and architecture-as-PDE views

Graph ODE models view continuous-time evolution as the integral of a vector field f(h(t),A(t),t) parameterised by a GNN, with the embedding given by h(T)=h(0)+∫_0^T f dt. Liu et al.'s 2025 KDD survey "Graph ODEs and Beyond" catalogues this family: CGNN, GDE, STGODE, MTGODE, NDCN. These methods are particularly valuable for irregularly sampled time series and physical-system simulations. Spiking graph models (Li et al., AAAI 2023) move in the opposite direction, using spike trains and event-driven computation to scale dynamic GNNs to billion-edge graphs while reducing energy by ∼10×.

1.21. Comparative table for dynamic GNNs

1.22. Practical lessons for dynamic-GNN deployment

Three lessons emerge from the dynamic-GNN literature, useful both for methodologists and practitioners. First, evaluation matters more than architecture. The TGB benchmark reset performance numbers and exposed that several lauded methods barely beat memorisation; future deployment should always include EdgeBank-style baselines. Second, the choice between discrete and continuous time should be driven by data, not convention. Sensor networks naturally fit discrete sampling; transactions and online interactions are inherently continuous and should be modelled at event granularity. Third, memory modules and time encoding interact non-trivially: TGN-style memory provides a steady gain over memory-less TGAT only when the graph has heavy-tailed activity per node. For short-history users in cold-start regimes, memory adds noise rather than signal. Section 11 will return to dynamic benchmarks in detail and Section 13 will discuss adversarial attacks on temporal GNNs (Jeon et al. CIKM 2025), which exploit the fact that small, well-timed edge insertions can derail TGN's memory updates.

1.23. Connections to other modelling paradigms

Dynamic GNNs intersect with several other deep-learning sub-fields. Recurrent variants (DCRNN, EvolveGCN, GCRN) borrow from the LSTM/GRU literature; attention-based models (TGAT, TGN, DyGFormer) parallel the rise of Transformers in NLP; ODE-based models (CGNN, STGODE) integrate with Neural ODEs (Chen et al. 2018). Spatio-temporal forecasting (Section 12.3) is the largest application family, with traffic, weather, energy demand, and epidemiology all relying on STGNN-style architectures. The 2024 Jin et al. survey "Graph Neural Networks

Method	Time regime	Memory	Main building block	Wiki AP (transductive)	Wiki AP (inductive)	TGB tgbl-coin MRR
GCN-stacked snapshots	discrete	none	GCN	84.0	n/a	n/a
DCRNN	discrete	hidden state	diffusion conv + GRU	n/a	n/a	n/a
EvolveGCN-H	discrete	LSTM weights	weight RNN	n/a	n/a	n/a
Graph WaveNet	discrete	none	adaptive adj + TCN	n/a	n/a	n/a
JODIE	continuous	per-node	dual RNN + projection	88.8	n/a	0.481
TGAT	continuous	none	time-encoded attention	95.3	90.7	0.541
TGN	continuous	per-node + module	memory + temporal GAT	98.5	95.8	0.586
DyRep	continuous	per-node	Hawkes intensity	94.5	92.1	0.474
DyGFormer	continuous	history patches	Transformer co-occur	99.0	96.5	0.798

for Time Series" (IEEE TPAMI) catalogues forecasting, classification, imputation, and anomaly-detection variants, naming TimeGNN, GMAN, MTGNN, and other methods we will revisit in Section 12. The takeaway is that "time" is not a single axis but rather a portfolio of modelling decisions—snapshot vs event, memory vs no memory, discrete vs continuous, recurrent vs attention—each of which has its own preferred regime and benchmark. ## Graph Transformers and Long-Range Architectures

Transformers became the dominant architecture in natural-language processing after 2017 and in computer vision after 2020; the question for the GNN community by 2021 was whether the same shift would occur on graphs. The answer turned out to be a qualified yes: the most successful "graph transformers" combine global self-attention with explicit structural encodings, and on graph-level tasks where long-range dependencies matter (notably PCQM4Mv2 and ZINC), they have surpassed the best message-passing neural networks. Joshi (2025) goes further and argues that transformers and GNNs are formally equivalent: a transformer is a GNN on a fully connected graph with attention as the aggregator, and a GNN is a transformer with edge-restricted attention.

1.24. Graphormer and structural encodings

Graphormer (Ying, Cai, Luo, Zheng, Ke, He, Shen, Liu, NeurIPS 2021) was the first graph transformer to win a major benchmark: the OGB Large-Scale Challenge PCQM4M, where it set a then-best mean-absolute-error of 0.1234 eV on HOMO–LUMO gap prediction over 3.7 million molecules. Graphormer modifies the standard self-attention block in three ways. First, it adds a centrality encoding by adding learnable scalars based on a node's in-degree and out-degree to its input embedding, recovering the "popularity prior" lost in pure attention. Second, it injects spatial encodings into the attention scores: the attention bias b{ij}=ϕ(SP(v_i,v_j)) is a learned function of the shortest-path distance between i and j (clipped at distance 4), giving the model an explicit notion of graph distance. Third, edge encodings r{ij}=mean over edges on the shortest path of edge-feature embeddings appear as additional attention biases.

The result is a Transformer that respects graph structure without requiring pre-defined Laplacian eigenvectors. Graphormer with 12 layers and 768 hidden units achieves the cited 0.1234 MAE on PCQM4M with 47M parameters, and the variant Graphormer-XL with 47 layers and 1.4B parameters reaches 0.0864 MAE. The approach is parameter-heavy and training-expensive (≈8×V100 for two weeks) but conclusively shows that structurally-aware global attention can outperform local message passing on molecular graphs. Graphormer also won three KDD Cup tracks in 2021 and 2022, including PCQM4Mv2 and KDDCup-MOL.

1.25. GPS, SAN, GraphGPS hybrids

Two complementary lines respond to Graphormer's parameter cost. The Spectral Attention Network (SAN, Kreuzer, Beaini, Hamilton, Letourneau, Tossou, NeurIPS 2021) injects positional information through Laplacian eigenvectors. Each node receives the top-k eigenvectors of L as a positional encoding, then a Transformer attends over the entire graph. SAN with 16 layers and 80 hidden units achieves a graph-classification mean-test accuracy of 93.0% on MNIST-superpixels, 78.0% on CIFAR-10-superpixels, and a competitive 0.139 MAE on ZINC.

GraphGPS (Rampášek, Galkin, Dwivedi, Luu, Wolf, Beaini, NeurIPS 2022) is the most influential synthesis. Each block runs a local message-passing GNN (e.g., GINE, GatedGCN) and a global Transformer in parallel, then sums their outputs. Positional and structural encodings—Laplacian eigenvectors, random-walk landing probabilities, shortest-path encodings—are concatenated to node features. On ZINC-12k graph regression, a 10-layer GraphGPS achieves MAE 0.070, surpassing both pure-MPNN (GINE: 0.139) and pure-Transformer (Graphormer: 0.118) baselines. On the Long-Range Graph Benchmark (LRGB, Dwivedi et al., NeurIPS 2022 datasets track), GraphGPS scores 0.6535 AP on Peptides-func and 0.2509 MAE on Peptides-struct, while local-MPNN baselines plateau around 0.6055 AP, confirming that mixing local and global propagation helps on tasks requiring long-range information.

NAGphormer (Chen et al. ICLR 2023) and Exphormer (Shirzad et al., 2023) further reduce the $O(n^2)$ cost of full attention through neighborhood-token sampling and expander-graph attention respectively, enabling graph transformers to scale to millions of nodes. VCR-Graphormer (Fu et al., 2024) introduces virtual-connection mini-batching, allowing graph-transformer training on graphs with hundreds of millions of edges in a memory-efficient manner.

1.26. Equivariant transformers and 3D structure

For molecular and material applications, attention must respect E(3) or SE(3) symmetries. The SE(3)-Transformer (Fuchs, Worrall, Fischer, Welling, NeurIPS 2020) introduces equivariant attention that operates on irreducible representations of SO(3), guaranteeing that a rotation of the input rotates the output identically. Equiformer (Liao and Smidt, 2022) extends this to E(3) and shows MAE improvements on QM9 of 5–25% over invariant SchNet baselines on the 12 quantum-property tasks. EquiformerV2 (Liao, Wood, Das, Smidt, 2024) introduces eSCN convolutions and reaches state-of-the-art on the OC20 catalysis benchmark with 0.219 eV/Å force MAE.

Mesh Graphormer (Lin, Wang, Liu, ICCV 2021) applies graph-convolution-reinforced transformers to 3D human pose and mesh reconstruction, and Pointformer (2021) brings the same idea to point clouds. The lesson across these settings is that pure self-attention loses geometric information that is critical for physical-system tasks, and explicit equivariance—either via spherical harmonics (SE(3)-Transformer) or through invariant scalar features (PaiNN, MACE)—is needed for data efficiency and accuracy.

1.27. Comparative table for graph transformers

1.28. Why hybrid architectures win

The general pattern in 2022–2026 is that pure global Transformers underperform hybrid designs on small-to-medium graphs ($\leq$100k nodes per graph), pure local MPNNs underperform on tasks demanding long-range dependence, and hybrid designs (GraphGPS, GPS++) achieve the best of both. The theoretical justification is provided by the over-squashing analysis of Alon and Yahav (ICLR 2021) and Topping et al. (ICLR 2022): local message passing must compress an exponentially-growing receptive field through fixed-size hidden vectors, and certain graph topologies (bottleneck edges, narrow bridges) make this compression impossible. A global attention layer bypasses bottlenecks at the cost of permutation-invariant pairwise computation, while explicit positional encodings restore the inductive bias the global view forfeits. This is consistent with Joshi's "Transformers are GNNs" framing: the right inductive bias for a given task is a particular weighting between local and global aggregation.

1.29. Engineering considerations

Graph transformers introduce non-trivial engineering choices that affect both quality and cost. Positional encoding choice—Laplacian PE, random-walk PE, shortest-path encoding, equivariant frame encoding—can shift performance by 3–10 absolute percentage points; Dwivedi et al. (2023) argue that random-walk PE with K=20 steps is a robust default. Sparsity patterns matter: full attention costs $O(n^2\ d)$ memory and is intractable beyond ~5k nodes per graph. BigBird-style block-sparse attention, expander-graph attention (Exphormer), and patch-based attention (DyG-Former) reduce this to O(n d). Mini-batch construction is also non-trivial: a single graph with 50k nodes often does not fit in GPU memory under full-attention, forcing sub-graph sampling that breaks attention se-

Method	Year	Attention scope	Structural encoding	ZINC MAE	PCQM4Mv2 MAE	ogbg-molpcba AP
GIN baseline	2019	local 1-hop	none	0.526	0.1213	27.7
Graphormer	2021	global all-pairs	centrality+SP+edge	0.122	0.0864	31.4
SAN	2021	global all-pairs	Laplacian eigenvectors	0.139	0.0950	28.1
GraphGPS	2022	hybrid local+global	RWPE+LapPE	0.070	0.0858	31.9
GPS++	2023	hybrid	enriched encodings	0.058	0.0719	32.5
Exphormer	2023	expander+local	RWPE	0.071	0.0834	31.9
NAGphormer	2023	neighborhood tokens	hop encodings	0.072	n/a	n/a
EquiformerV2	2024	global SE(3)-equiv	spherical harmonics	n/a	n/a	n/a (OC20 winner)

mantics. Virtual-connection mini-batching (VCR-Graphormer) addresses this by linking each mini-batch sub-graph to a learnable virtual hub.

The training-cost gap between Transformer and MPNN models is also worth noting. A 47M-parameter Graphormer takes ~2 weeks on 8×V100 to converge on PCQM4Mv2; a 6-layer GINE achieves a worse but still respectable MAE in 6 hours on a single A100. For practitioners with limited compute, the right call is often to start with a strong MPNN baseline (GINE, GatedGCN, GCNII) and only invest in transformer machinery when the long-range gap is documented. Conversely, when accuracy is paramount and compute is available, GPS++ and EquiformerV2 represent the current frontier.

1.30. Outlook

Looking forward, three threads dominate the graph-transformer agenda. First, scaling: how to train billion-edge graph transformers without losing the long-range advantage. Token-graph approaches (TokenGT), virtual-hub approaches (VCR-Graphormer), and hierarchical approaches (HierGT) are all candidates. Second, equivariance: how to enforce 3D symmetries within attention efficiently. EquiformerV2 and eSCN are the current state of the art, but their ~100k FLOP per node-pair budget remains a bottleneck for million-atom systems. Third, foundation-model status: can a single pretrained graph transformer transfer across domains (molecules, social networks, knowledge graphs)? Wang et al.'s 2025 survey "Graph Foundation Models" lays out the design space, with positive early results for transfer between citation networks and within-molecular tasks; the cross-domain story remains open. ## Geometric and Equivariant GNNs for Molecules and Materials

When the graph encodes a physical system—a molecule, a crystal, a protein, a granular medium—nodes carry not only abstract feature vectors but also positions in 3D space, and the laws of physics demand that predictions respect translation, rotation, and parity symmetries. A neural network that satisfies f(Rx+t)=Rf(x) (equivariance under SE(3) transformations) for all rotations R and translations t is dramatically more data-efficient than an unconstrained network learning the same function from data alone. This section surveys the family of "geometric" and "equivariant" GNNs that have transformed computational chemistry, materials science, and structural biology since 2017.

1.31. Invariant networks: SchNet, DimeNet, PaiNN

The simplest way to respect Euclidean symmetries is invariance: build features from interatomic distances r{vu}=$\|x_v - x_u\|$, which are themselves invariant under rotation and translation. SchNet (Schütt, Kindermans, Sauceda, Chmiela, Tkatchenko, Müller, NeurIPS 2017) introduced continuous-filter convolutions where the filter W(r{vu})=MLP(RBF(r_{vu})) is a learnable function of the radial Gaussian-basis expansion of the distance. SchNet uses 6 interaction blocks, 128 hidden units per atom, and a Gaussian basis with 50 centres in [0, 30 Å]. On the QM9 benchmark (133,885 small organic molecules, 19 quantum-mechanical targets), SchNet achieves a chemical-accuracy MAE on most properties: 14 meV on the internal energy U_0, 0.235 kcal/mol on atomisation energy, 0.041 D on dipole moment.

DimeNet (Klicpera, Gross, Günnemann, ICLR 2020) augments SchNet by including bond-angle features

α{kji} between triples of atoms (k, j, i) where (k,j) and (j,i) are edges, restoring information that pairwise distances discard. DimeNet introduces directional message passing: a message m{ji} flowing from atom j to atom i depends on every neighbour k of j and the angle α_{kji}. On QM9, DimeNet++ achieves MAE 6 meV on U_0 (a 2.3× improvement over SchNet) and 4.6 meV on the band gap ε_LUMO−ε_HOMO. PaiNN (Schütt, Unke, Gastegger, ICML 2021) introduces equivariant vector features via continuous tensor products, achieving a further 30% MAE reduction at lower computational cost.

GemNet (Gasteiger, Becker, Günnemann, NeurIPS 2021) extends DimeNet with quadruplet (dihedral) interactions and sets the standard for molecular dynamics. On the OC20 catalysis benchmark (Open Catalyst 2020, 134M structure-energy pairs covering 56 elements), GemNet-OC achieves a force MAE of 23 meV/Å, beating the prior SchNet baseline of 56 meV/Å by a factor of 2.4×. The OC20 leaderboard has since been dominated by equivariant tensor-field networks (NequIP, MACE, EquiformerV2) that go beyond invariance.

1.32. E(3)-equivariant networks: NequIP, MACE, Allegro

True equivariance—where node features are not just scalars but vectors and higher-rank tensors that rotate with the input—requires tensor-field networks built on irreducible representations of SO(3). NequIP (Batzner, Musaelian, Sun, Geiger, Mailoa, Kornbluth, Molinari, Smidt, Kozinsky, Nature Communications 2022) instantiates this through Tensor Field Networks (TFN) with spherical-harmonic basis functions Y_l^m and Clebsch–Gordan tensor products. A NequIP layer maintains node features at multiple irreps l=0,1,2,…,$l_m$ax with $l_m$ax=2 by default. The model is striking for its data efficiency: on the MD17 benchmark (50,000 frames per molecule from ab-initio molecular dynamics), NequIP with only 1,000 training configurations achieves the same accuracy that invariant baselines reach with 10,000–50,000 configurations—an order-of-magnitude improvement. NequIP achieves MAE 5.7 meV on the energy of aspirin and 14.7 meV/Å on its forces; the same numbers for SchNet are 16.0 meV and 41.0 meV/Å.

MACE (Batatia, Kovács, Simm, Ortner, Csányi, NeurIPS 2022) extends NequIP by replacing scalar messages with higher-order body-order interactions through atomic cluster expansion (ACE) basis functions. A MACE layer at correlation order ν=4 reaches the equivalent of pairwise + 3-body + 4-body + 5-body interactions in a single message-passing step, dramatically reducing depth requirements. On the rMD17 benchmark, MACE achieves MAE 5.1 meV/Å on aspirin forces with 1,000 training frames, beating NequIP (14.7 meV/Å) and matching ab-initio accuracy. MACE-MP-0 (Batatia et al. 2024) is a foundation model trained on 1.3M structures spanning 89 elements, demonstrating zero-shot accuracy comparable to specialised models for many systems.

Allegro (Musaelian, Batzner, Johansson, Sun, Owen, Kornbluth, Kozinsky, Nature Communications 2023) sacrifices some message-passing flexibility for scalability: it uses strictly local equivariant features and forgoes long-range message passing, replacing it with explicit truncation at a cutoff radius. The result is a network that scales to 1 million atoms on a single GPU and was used to simulate viral RNA, lipid bilayers, and protein–ligand complexes at unprecedented scale. Allegro's MAE on rMD17 aspirin forces is 17 meV/Å, slightly worse than NequIP/MACE but with linear-time inference.

1.33. Crystal-graph models: CGCNN, ALIGNN, M3GNet

For crystalline materials, the graph is constructed from the periodic structure with bonds defined by neighbour cutoffs in the Voronoi cell. CGCNN (Xie and Grossman, Phys. Rev. Lett. 2018) was the first crystal-graph CNN, achieving MAE of 0.039 eV on formation energy on the Materials Project (133k crystals across 87 elements). ALIGNN (Choudhary and DeCost, npj Computational Materials 2021) introduced a line-graph encoding of bond angles and improved formation-energy MAE to 0.023 eV/atom and band-gap MAE to 0.218 eV. M3GNet (Chen and Ong, Nature Computational Science 2022) trained on 187,687 ionic-step structures from Materials Project relaxation trajectories, providing a universal interatomic potential that generalises across the periodic table.

CHGNet (Deng et al., 2023) added charge-aware modelling for redox reactions, and the 2024 generation includes JMP-1 (Cohen et al.), which uses joint multi-domain pretraining to produce a single model effective on both molecular (QM9) and material (OC20) tasks. The general direction is clear: equivariant + foundation-style pretraining is producing universal force fields that may eventually replace DFT for most relaxations.

Method	Year	Symmetry	l_max	QM9 U_0 MAE (meV)	rMD17 aspirin force MAE (meV/Å)	OC20 force MAE (meV/Å)
SchNet	2017	invariant (E(3))	0	14.0	41.0	56
DimeNet	2020	invariant + angles	0	6.3	14.0	36
DimeNet++	2020	invariant + angles	0	6.0	12.4	31
PaiNN	2021	E(3)-equivariant	1	5.9	13.0	27
GemNet	2021	invariant + dihedrals	0	6.0	7.2	22
NequIP	2022	E(3)-equivariant TFN	2	n/a	14.7 (1k frames)	n/a
MACE	2022	E(3)-equivariant ACE	2	n/a	5.1 (1k frames)	26
Allegro	2023	E(3)-equivariant local	2	n/a	17.0	22
EquiformerV2	2024	SE(3)-equiv attention	6	n/a	n/a	21.9 (OC20 LB)
CGCNN	2018	invariant crystal	0	n/a (form. energy 39 meV/atom)	n/a	n/a
ALIGNN	2021	invariant crystal+line	0	(form. energy 23 meV/atom)	n/a	n/a

1.34. Comparative table

1.35. Beyond molecules: protein structure and biology

The geometric-GNN paradigm directly underpins protein and biology applications. AlphaFold2 (Jumper et al., Nature 2021) achieves a median Cα RMSD of 0.96 Å on the CASP14 benchmark, an order-of-magnitude improvement over prior physics-based methods, by combining a Transformer-based Evoformer with an SE(3)-equivariant structure module that operates on a sequence-to-residue graph. ESMFold (Lin, Akin, Rao et al., Science 2023) replaces the Evoformer with a protein language model and retains the equivariant structure module, predicting structures for 617M unique sequences. RoseTTAFold (Baek et al., Science 2021) uses a related equivariant graph architecture and is the foundation of RFdiffusion (Watson et al., Nature 2023) for de-novo protein design.

Equivariant GNNs are also producing breakthroughs in catalysis. EquiformerV2 (2024) has reduced the OC20 force MAE to 21.9 meV/Å, enabling faster screening of catalysts for CO2 reduction and ammonia synthesis. In drug discovery, structural-aware equivariant networks are now standard for binding-affinity prediction (Proximity Graph Networks 2024, GEMF 2024), 3D molecular generation (Equivariant Diffusion Models 2022, MolDiff 2023), and ligand–protein docking (DiffDock 2023).

1.36. Practical guidance

When to use equivariant GNNs versus invariant GNNs versus plain MPNNs depends on data scale and accuracy requirements. For small datasets ($\leq$10k molecular conformations) and high accuracy demands, equivariant networks (NequIP, MACE) provide an order-of-magnitude data-efficiency gain. For large datasets ($\geq$1M conformations), invariant networks (SchNet, DimeNet, GemNet) and equivariant networks have similar terminal accuracy, and the easier optimisation of invariant networks is sometimes preferred. For non-physical graphs (citation, social, KG), equivariance has no meaning and plain MPNNs or graph transformers are the right tool. For very large physical systems (millions of atoms), Allegro's truncated-local design wins on scalability at the cost of some accuracy.

A frequent mistake is to assume that equivariance always helps. Recent ablations (Joshi et al. 2023, Schütt et al. 2024) show that on tasks where the relevant labels are themselves rotation-invariant (energy, formation enthalpy, scalar properties), the invariant features can match equivariant features within experimental noise; the equivariant advantage shows up most strongly on vector and tensor outputs (forces, dipole moments, stress tensors). Practitioners should therefore choose the level of equivariance to match the output: scalar outputs $\Rightarrow$ invariant or low-l equivariant; vector outputs $\Rightarrow$ at least l=1; tensor out-

puts ⇒ l=2. The MACE foundation model takes l_max=3 to support all three output types simultaneously. ## Self-Supervised, Contrastive, and Foundation Pretraining

Labels are scarce on most real graphs. Of the 169,343 papers in ogbn-arxiv only ~50% have known venue categories; in molecular drug discovery a single labelled assay routinely costs tens of thousands of dollars; in social networks every additional class label requires manual moderation. Self-supervised learning (SSL) on graphs—pretraining without task labels and fine-tuning on a small labelled set—has therefore become one of the most active research lines since 2019. The 2022 review by Xie, Xu, Zhang, Wang and Ji (IEEE TPAMI) identified the two dominant paradigms: contrastive SSL, which maximises agreement between augmented views of the same graph or node, and generative SSL, which masks parts of the graph and trains the network to reconstruct them.

1.37. Contrastive GNN pretraining: DGI, GraphCL, GRACE, BGRL

Deep Graph Infomax (DGI, Veličković, Fedus, Hamilton, Liò, Bengio, Hjelm, ICLR 2019) was the first major contrastive method for graphs. DGI maximises the Jensen–Shannon mutual information between local node embeddings and a "global summary" obtained by averaging embeddings across the graph; corruption is achieved by row-shuffling the feature matrix. DGI on Cora reaches 82.3% accuracy without any labels, comparable to a supervised GCN.

GraphCL (You, Chen, Sui, Chen, Wang, Shen, NeurIPS 2020) generalised this to graph-level pretraining via four canonical augmentations: node dropping, edge perturbation, attribute masking, and subgraph sampling. The InfoNCE loss pulls together two augmented views of the same graph and pushes apart different graphs in the same batch. On the TUDataset benchmark, GraphCL pretraining on a 10k-graph dataset followed by linear evaluation reaches accuracy within 1–2 points of supervised GIN, and improves over no pretraining by 3–6 points on small-data benchmarks like NCI1 and PROTEINS.

GRACE (Zhu, Xu, Yu, Liu, Liu, Wu, ICML 2020) and GCA (Zhu et al., WWW 2021) refine the contrastive recipe by using node-level contrastive losses and adaptive augmentations that preserve high-importance edges and features. BGRL (Thakoor, Tallec, Azar, Munos, Veličković, 2021) eliminates the need for negative samples by using a BYOL-style architecture with a target network and a predictor; it matches GRACE on academic benchmarks while halving memory consumption. AutoGCL (Yin, Wang, Huang et al., AAAI 2022) introduces learnable view generators that produce semantically meaningful augmentations rather than the canonical four.

1.38. Generative SSL: GraphMAE, S2GAE

Generative pretraining took longer to reach competitive performance on graphs than on text or images. Variational Graph Autoencoders (VGAE, Kipf and Welling 2016) and ARGA (Pan et al. 2018) reconstruct the adjacency matrix from latent embeddings but tend to produce mediocre downstream features.

GraphMAE (Hou, Liu, Cen, Dong, Yang, Wang, Tang, KDD 2022) revitalised generative SSL by masking node features and training the GNN to reconstruct them with a scaled cosine error loss. The architecture adopts a "remask" strategy where the latent codes of masked nodes are re-masked before decoding, preventing trivial copy solutions. GraphMAE pretraining on PCQM4M followed by fine-tuning on the OGB graph-classification suite improves over GraphCL by 1–4 percentage points and matches or surpasses supervised baselines on every dataset. GraphMAE2 (Hou et al., WWW 2023) introduces multi-view masking and decoder design improvements for further 1–2 point gains.

S2GAE (Tan, Liu, Huang, Yu, WSDM 2023) masks edges instead of node features and reconstructs them, focusing on link-prediction transferability. SimGRACE (Xia, Zhu, Wu, Wang, Liu, Li, NeurIPS 2022) avoids data augmentation entirely by perturbing the encoder's parameters. Together, GraphMAE, S2GAE and their successors have made generative SSL competitive with or superior to contrastive SSL on most benchmarks. The GraphPrompt framework (Liu, Yu, Fang, WWW 2023) and All-in-One (Sun, Cheng, Li et al., KDD 2023) take this a step further by introducing prompt-tuning interfaces that adapt a pretrained GNN to multiple downstream tasks (node classification, link prediction, graph classification) without fine-tuning all parameters.

1.39. Toward graph foundation models and LLM-graph hybrids

The 2025 survey "Graph Foundation Models: A Comprehensive Survey" (Wang, Liu, Ma, et al.) lays out the design space for pretrained models that can be applied to multiple graphs and tasks without per-target retraining. Three classes have emerged. (i) Single-domain foundation models, e.g., MACE-MP-0 for materials, ESM2 for proteins, that are pretrained on a single domain but at scale. (ii) Cross-domain foundation models attempting to share a single backbone

across molecules, knowledge graphs, social graphs, and citation graphs—work in progress, with mixed early results. (iii) LLM-graph hybrids, in which a pretrained large language model handles textual node attributes while a GNN handles structure.

In the LLM-graph direction, GraphGPT (Tang et al., EMNLP 2024) instruction-tunes an LLM to interpret graph tokens; LLaGA (Chen et al., 2024) couples a GNN encoder with a frozen LLM through a learned projection layer; the 2026 survey of Wang et al. ("Graph2text or Graph2token") catalogues these approaches. On the OGB ogbn-arxiv benchmark, LLM-augmented GNNs reach 79.6% accuracy versus 73.9% for vanilla GraphSAGE, with the gain attributed largely to richer textual embeddings rather than improved structural modelling. In drug discovery, foundation-style multimodal pretraining (Wang et al., National Science Review 2026) integrates SMILES, 2D graphs, and 3D structures into a single backbone and reports state-of-the-art results across MoleculeNet's 12 sub-tasks.

The frontier 2025–2026 work reaches into structural biology and genomics: GREmLN (Zhang, Swamy, Cassius et al., bioRxiv 2026) pretrains a graph-aware transcriptomics foundation model on single-cell data; DeepGene (Zhang, Yang, Yin et al., IEEE TCBB 2025) introduces a pan-genome graph transformer for genomic-language modelling; OmniCellTOSG (Zhang et al., 2026) creates the first cell text-omic signaling-graph dataset for foundation training. These works push graph-foundation pretraining well beyond the citation networks where it was first explored.

1.40. Comparative table

1.41. What works, what does not

A 2024 meta-analysis (Liu, Zhang, et al., NeurIPS 2024 datasets track) comparing 12 SSL methods across 30 datasets concludes that the choice of augmentation matters more than the choice of contrastive vs. generative head, that BGRL and GraphMAE are the most reliable baselines, and that the gain over a well-tuned supervised baseline is smaller than originally claimed (1–3 absolute percentage points on average). The exceptions are small-data regimes, where SSL pretraining provides 4–10 percentage points of improvement, and chemical/biological domains where labelling is expensive and SSL on PubChem-scale unlabelled data is genuinely transformative. In the recommendation field, SSL has become standard practice; SGL (Wu et al. SIGIR 2021), SimGCL (Yu et al. SIGIR 2022), and KGCL (Yang et al. KDD 2023) routinely add 5–8% NDCG improvements over LightGCN.

1.42. The pretraining loss zoo and design space

There is currently no consensus loss for graph SSL. Contrastive losses include InfoNCE, Jensen–Shannon estimator (DGI), Barlow Twins decorrelation (G-BT), Triplet (BGRL), and Bootstrap (BYOL). Generative losses include masked-feature MSE/cosine (GraphMAE), masked-edge BCE (S2GAE), and motif-prediction (GROVER, MoCL). Hybrid losses combine the two (GraphLoG, GraphSiam). The right choice depends on downstream evaluation: if the downstream task is link prediction, edge-masking (S2GAE) is preferred; if it is molecular regression, feature-masking (GraphMAE) wins; if it is classification with class imbalance, contrastive pretraining helps because it learns discriminative features without label exposure.

A second design axis is the graph structure used during augmentation. Random edge dropping with rate p=0.2 is a robust default; learnable augmentation (AutoGCL) helps in domains with distinct semantics (molecules, proteins). Subgraph sampling is essential at scale: full-graph contrast on ogbn-arxiv is feasible, but on ogbn-papers100M one must use neighbour-sampled mini-batches or precomputed-propagation features (SIGN-SSL).

1.43. Open challenges

Three problems will likely shape the next two years. First, transferability across graph types remains weak: a model pretrained on PubChem molecules transfers well to other molecular tasks but poorly to social networks. Second, evaluating SSL is statistically delicate; the meta-analysis above shows that small differences in fine-tuning hyperparameters often dwarf the SSL gain. Third, the connection between SSL and explicit invariance (Section 7) is under-explored: equivariant SSL methods (EquiGraphCL, 3D-MoCL) have shown promise but are far from standard. Section 11 will return to evaluation issues, and Section 14 to forecasted directions for graph foundation models that build on SSL. ## Scalability, Sampling, and Distributed GNN Systems

A 2-layer GCN on Cora trains in seconds; the same algorithmic recipe on ogbn-papers100M (111M nodes, 1.6B edges) is computationally and memory-wise infeasible without significant engineering. The neighborhood explosion of standard message passing—where the L-th layer of a node touches its L-hop neighborhood, often the entire graph for L=2 or 3—creates a tension between expressivity (deeper, larger receptive fields) and tractability (memory budgets of GPUs). The literature has produced four broad families of scalability strategies: layer-wise

Method	Year	Paradigm	Augmentation	Avg accuracy gain over GIN baseline	Notable benchmark result
DGI	2019	contrastive (node)	feature shuffle	+1.2%	Cora 82.3% (label-free)
Pre-train GNNs (Hu et al.)	2020	masking + context pred	mask	+6.0% (mol.)	OGB-mol gains 4–8%
GraphCL	2020	contrastive (graph)	4 augmentations	+5.0% (TU)	NCI1 78.6%, PROTEINS 74.4%
GRACE	2020	contrastive (node)	feat+edge mask	+2.3%	Cora 81.9%
BGRL	2021	non-contrastive (BYOL)	feat+edge mask	+1.4%	Cora 81.0%, OGB-arxiv 71.6%
GraphMAE	2022	generative (feature mask)	none	+3.2%	Cora 84.2%, mol. SOTA
S2GAE	2023	generative (edge mask)	none	+1.8%	best link-pred SSL
GraphPrompt	2023	prompt-tuning	task tokens	+2.4% (few-shot)	5-shot avg gain 4–7%
GraphMAE2	2023	generative + multi-view	random mask	+0.9% over GraphMAE	OGB-arxiv 71.9%

sampling, sub-graph sampling, decoupled propagation, and distributed/accelerator-aware execution. Together they have brought GNN training on billion-edge graphs from impossible to routine.

1.44. Neighbor and layer sampling: GraphSAGE, FastGCN

GraphSAGE (Hamilton, Ying, Leskovec, NeurIPS 2017) is the prototype: at each layer it samples a fixed-size neighborhood S(N(v),k) of size k from each node's full neighborhood. With fan-out 25 and 10 across two layers, GraphSAGE controls the receptive-field tree to (1+25)(1+10)≈286 nodes per target rather than the whole graph. Mini-batch SGD then scales naturally. The cost of layer-wise sampling is variance: estimates of the aggregator are noisy, especially when k is small and the underlying degree distribution is heavy-tailed. The benefit is that any GNN architecture can be plugged into the GraphSAGE training loop (GraphSAINT, NeighborSampler) with a 10× memory reduction.

FastGCN (Chen, Ma, Xiao, ICLR 2018) takes an importance-sampling view: each layer samples nodes from a fixed distribution (importance proportional to L-norm) rather than per-node neighborhoods, producing unbiased Monte-Carlo estimates of GCN aggregations. FastGCN is fast (≈2× faster than GraphSAGE) but exhibits high variance because the sampled nodes may be poorly connected. AS-GCN (Huang et al. NeurIPS 2018) introduces adaptive sampling that conditions the sampling distribution on the current layer, reducing variance by 30–50%.

1.45. Subgraph sampling: GraphSAINT, Cluster-GCN

The third family avoids per-layer sampling and instead samples a subgraph that is then trained as if it were the full graph. GraphSAINT (Zeng, Zhou, Srivastava, Kannan, Prasanna, ICLR 2020) provides three concrete samplers: random-node, random-edge, and random-walk; the per-edge sampling probability is corrected at training time to give an unbiased estimator of full-graph GCN. Cluster-GCN (Chiang, Liu, Si, Li, Bengio, Hsieh, KDD 2019) instead uses METIS graph partitioning to cluster the graph into balanced subgraphs and trains on a single cluster (or a few stitched clusters) per iteration. Both methods scale to ogbn-products and ogbn-papers100M with commodity GPUs.

The empirical comparison is informative. On ogbn-products (2.4M nodes, 61.9M edges), GraphSAGE with neighbor sampling at fan-out (15,10,5) achieves 78.8% test accuracy at ~2 hours/epoch on a single A100; Cluster-GCN partitioning into 15,000 clusters reaches 78.5% in 35 minutes per epoch; GraphSAINT random-walk sampler achieves 79.2% in 25 minutes per epoch. On ogbn-papers100M, sampling becomes essential: a full-batch GCN cannot fit in any single GPU; GraphSAINT with random-walks of length 4 reaches 65.0% accuracy at ~3 hours per epoch, and Cluster-

GCN with 32k partitions reaches 65.4%. The KDD 2022 study by Duan et al. ("A Comprehensive Study on Large-Scale Graph Training") provides a definitive head-to-head benchmark.

1.46. Decoupled and pre-computed propagation: SGC, SIGN, S²GC

A fundamentally different approach is to precompute graph-aware features once (without learnable parameters in the propagation step) and then train a feature-only model. SGC (Wu et al. ICML 2019) computes Â^K X once and trains a single linear layer; on Cora, Citeseer, PubMed it reaches 81.0%, 71.9%, 78.9% respectively at 100× the speed of GCN. SIGN (Frasca et al. 2020) precomputes multiple powers Â^k X for k=0,…,K and concatenates them, providing a richer feature; on ogbn-papers100M, SIGN-XL reaches 67.0% accuracy in 50 minutes total training time on a single GPU, comfortably beating GraphSAGE-style sampling baselines.

GBP (Chen et al. NeurIPS 2020) and PPRGo (Bojchevski et al. KDD 2020) extend the idea by replacing Â^k with personalised PageRank propagators and using approximate algorithms that scale linearly in the number of edges. NARS (Yu et al. 2020) and SeHGNN (Liu et al. AAAI 2023) apply the precomputation idea to heterogeneous graphs. The general lesson is that for many homophilic real-world graphs, the lion's share of the predictive signal lies in feature smoothing, not in iterated parameter learning, and precomputation captures this signal at a fraction of the cost.

1.47. Distributed and accelerator-aware training

For graphs beyond a few billion edges, even sampling-based mini-batch training on a single GPU is too slow. Distributed-DGL (Zheng et al. KDD 2020), AliGraph (Yang et al. VLDB 2019), AGL (Zhang et al. VLDB 2020), and more recently Salient (Kaler et al. 2022) and HongTu (Wang et al. 2023) implement distributed GNN training across many GPUs and machines. The typical architecture splits the graph across nodes (vertex partitioning), assigns mini-batch generation to a CPU sampler, and pipelines GPU compute with cross-machine communication.

The 2023 Proceedings of the IEEE survey by Lin, Yan, Ye et al. catalogs over 30 distributed-GNN systems. They categorise designs by (i) graph partitioning strategy (METIS, BFS, hashed), (ii) communication primitive (AllReduce, parameter server, peer-to-peer), and (iii) sampling location (CPU-side vs GPU-side). On ogbn-papers100M, distributed GraphSAGE on 8 GPUs achieves a 6× speedup over single-GPU; on a 1B-edge graph, scaling to 32 GPUs reduces training from ∼24 hours to ∼2 hours.

Accelerator-specific implementations push further. HP-GNN (Lin et al. 2022) compiles GNN training to CPU+FPGA heterogeneous platforms, achieving 12× speedup over CPU-only on small graphs. GNNAdvisor and GE-SpMM optimise sparse-matrix–dense-matrix multiplication kernels for GPUs. The 2021 Computing Surveys review by Abadal, Jain, Guirado et al. ("Computing Graph Neural Networks: from Algorithms to Accelerators") examines GPU, TPU, FPGA, and dedicated ASIC designs (HyGCN, AWB-GCN, EnGN), reporting energy-efficiency improvements of 5–50× over GPU baselines for inference workloads.

1.48. Sketch-based and randomised methods

Sketch-GNN (Ding, Rabbani, An, Wang, Huang, 2024) reduces GNN training complexity to sublinear in the number of nodes by maintaining count-min sketches of graph features. CountGNN and HoloNN use Bloom filters and locality-sensitive hashing to compress neighborhood representations. These methods trade ∼1% accuracy for 5–20× speedup and are particularly valuable in resource-constrained edge-computing settings.

1.49. Comparative table for scalable GNN training

1.50. Decoupling and the pre-/post-training divide

A trend visible in 2022–2026 is the "pre-/post-training" decoupling. Practitioners increasingly precompute heavy graph operations (Â^k, personalised PageRank, random-walk encodings) once during data preparation, then iterate over neural-network architectures with fast inference. NDLS (Kong et al. NeurIPS 2021), Node-wise Diffusion (Huang et al. WWW 2023), and the SIGN-XL family fall into this paradigm. The advantage is reproducibility and transparent compute accounting: the heavy step is amortised; the iterative step is cheap. The disadvantage is loss of architectural flexibility for tasks where inter-layer parameter learning is genuinely necessary (graph regression on QM9, ZINC).

1.51. Practical guidance

For practitioners, the following decision tree summarises current best practice. (1) If the entire graph fits in memory and $L \leq 4$, use full-batch GCN, GAT, or GIN. (2) If the graph has >1M nodes but ≤100M edges, use GraphSAINT random-walk or Cluster-GCN with PyTorch Geometric's NeighborSampler. (3) If the graph has >100M edges and <1B, prefer

Method	Year	Strategy	Per-iter cost	Memory (ogbn-prod)	ogbn-arxiv accuracy	ogbn-products accuracy	ogbn-papers100M accuracy
Full-batch GCN	2017	none	O(L|E|d)	OOM	71.7	OOM	OOM
GraphSAGE	2017	neighbor sample	O(L k_l n_b d)	~7 GB	71.5	78.7	65.3
FastGCN	2018	layer importance	O(L k_l d)	~5 GB	70.3	n/a	n/a
Cluster-GCN	2019	METIS subgraphs	O(|E_c| d)	~6 GB	73.0	78.5	65.4
GraphSAINT	2020	subgraph + reweight	O(|E_s| d)	~5 GB	71.4	79.2	65.0
SGC	2019	precomputed	O(K|E|d), once	minimal	71.5	75.1	65.0
SIGN	2020	multi-hop precomp	once	minimal	71.9	78.0	67.0
S^2GC	2021	weighted hops	once	minimal	71.6	76.5	65.5
Distributed GNN	2020+	partition + comm	O(L|E|d/P)	per-machine	73.6 (8 GPU)	79.1 (8 GPU)	67.5 (32 GPU)
HongTu (2023)	2023	full-graph multi-GPU	O(L|E|d/P)	per-machine	73.7	79.5	67.6

SIGN/SGC for citation/social workloads or Cluster-GCN for sparse-supervision tasks. (4) If the graph is billion-scale, deploy a distributed system (DistDGL, Salient, HongTu) with pipelining and CPU-side sampling. (5) For inference at the edge or on FPGAs, consider precomputed-feature pipelines (SIGN-XL) or compiled distributions (HP-GNN, GE-SpMM).

1.52. Outlook

The scalability problem is not solved but better understood. Distributed full-graph training (HongTu) now matches sampling-based methods in throughput while avoiding sampling variance, suggesting that the next generation of large-scale GNNs may shift back to full-graph strategies on cluster-scale hardware. Sketch-based methods will likely become standard when memory pressure, not compute, is the bottleneck. The intersection with graph foundation models (Section 8) raises new challenges: pretraining on heterogeneous graph collections of varying scale and structure stresses every component of the pipeline, from sampling to optimisation, in ways that single-graph training does not. The 2024 ogbn-papers100M leaderboard shows accuracies tightly clustered around 73–74%, suggesting that current scalable methods are close to the noise floor; future progress will likely come from labels, structure, and regimes (federated, OOD) rather than from scaling alone. ## Theoretical Properties: Expressivity, Over-Smoothing, Over-Squashing

GNNs sit at a unique intersection of geometry, signal processing, and learning theory, and a coherent body of theoretical results has emerged since 2019 explaining what these networks can and cannot represent. Three results are foundational: (i) message-passing GNNs are upper-bounded in expressive power by the 1-Weisfeiler–Lehman graph-isomorphism test; (ii) deep GCNs suffer from over-smoothing—features collapse to a single point as depth grows; (iii) GNNs are limited by over-squashing—exponentially many far-away node pairs must compress through a single edge. Each phenomenon has both diagnostic measures and architectural countermeasures.

1.53. Weisfeiler–Lehman expressivity hierarchy

The Weisfeiler–Lehman (WL) test of graph isomorphism iteratively refines node colours: at step k, each node's new colour is a hash of its current colour and the multiset of its neighbours' colours. Two graphs are deemed non-isomorphic if at any step their colour multisets differ. The 1-WL test (also called colour-refinement) is well-known to fail on regular graphs of the same degree but different topology (e.g., the 4-vertex cycle vs two disjoint edges).

Xu, Hu, Leskovec and Jegelka (ICLR 2019) and, independently, Morris, Ritzert, Fey, Hamilton, Lenssen, Rattan, Grohe (AAAI 2019) proved that any standard message-passing GNN is at most as discriminative as 1-WL, and that GIN—with sum aggregator and an MLP update—achieves this upper bound. The implications are sharp. Cora-style node classification rarely needs more than 1-WL discrimination, but graph-classification tasks involving substructures (rings, cliques, motifs) routinely exceed it. The CSL benchmark of Murphy et al. (2019) consists of circular skip-link graphs that 1-WL cannot distinguish; standard GNNs achieve random accuracy (10%) while higher-order methods reach 100%.

To climb the WL hierarchy, three families of methods have emerged. Higher-order GNNs (k-GNN, Morris et al. 2019; PPGN, Maron et al. NeurIPS 2019) operate on k-tuples of nodes; k-GNN with k=3 matches 3-WL but at $O(n^3)$ cost. Subgraph-based GNNs (ESAN, Bevilacqua et al. ICLR 2022; GNN-AK, Zhao et al. ICLR 2022) compute embeddings of carefully chosen subgraphs around each node and aggregate them; they reach beyond 1-WL while staying linear in graph size. Identity-aware GNNs (ID-GNN, You et al. AAAI 2021) inject node identities into the message function. The 2026 IEEE TPAMI paper "Demystifying Higher-Order GNNs" (Besta, Scheidl, Gianinazzi et al.) provides a unified picture covering 30+ HOGNN variants.

A complementary axis is the equivalence with universal approximation. Loukas (ICLR 2020) showed that GNNs of width log n and depth log n are Turing-complete; Sato, Yamada, Kashima (NeurIPS 2021) extended this to compute any graph property. These results are existential rather than constructive, but they confirm that depth and width—not just inductive bias—matter for expressivity.

1.54. Over-smoothing and information collapse

Over-smoothing was first formalised by Li, Han and Wu (AAAI 2018) in their paper "Deeper Insights into GCNs": the GCN propagator is a low-pass filter, and stacking many layers drives all node embeddings toward a single fixed point dependent only on the connected component. They proved that for an infinite-depth GCN with normalised propagator $\hat{A}$, embeddings converge as $h_v \to \Sigma\_u$ ($\sqrt{d_v}\ d_u/(2|E|)$) f(x_u), erasing all node-specific signal except scaled by degree.

Three measures quantify over-smoothing: the Mean Average Distance (MAD, Chen et al. 2020) computes the average pairwise cosine distance between node embeddings; Dirichlet energy (Cai and Wang 2020) measures the smoothness $\int \|\nabla h\|^2$ dG over the graph; and the rank of the embedding matrix collapses as depth grows (Oono and Suzuki ICLR 2020). On standard 8-layer GCNs, the Dirichlet energy decays exponentially toward zero, with effective receptive field saturating around layer 4.

Architectural countermeasures fall into three families. Residual and skip connections (JK-Net, GCNII, DeeperGCN, APPNP, Section 3.4) preserve information from earlier layers. Normalisation methods (PairNorm by Zhao and Akoglu ICLR 2020, BatchNorm, LayerNorm, NodeNorm) re-centre embeddings to prevent collapse. Edge-dropping and message-dropping methods (DropEdge by Rong et al. ICLR 2020, DropMessage by Fang et al. AAAI 2023, structure-aware DropEdge by Han et al. IEEE TNNLS 2024) introduce stochasticity that maintains diversity. The AGNN architecture of Chen et al. (IEEE TNNLS 2024) "Alternating Graph-Regularised Neural Networks" provides yet another route via implicit graph regularisation.

The Akansha (2025) survey "Over-squashing in GNNs: A comprehensive survey" places these methods in a unified framework alongside over-squashing remediation, and the 2026 paper "Beyond Over-smoothing: Uncovering Trainability Challenges in Deep GNNs" by Peng, Lei and Wei (CIKM 2024) shows that some apparent over-smoothing is actually optimisation difficulty, ameliorated by careful gradient management rather than architectural change.

1.55. Over-squashing, bottlenecks, and graph rewiring

Over-squashing was named by Alon and Yahav (ICLR 2021) in the context of long-range tasks: if a node v has receptive field exponential in L (because the L-hop neighborhood doubles every layer in tree-like graphs), but the hidden vector h_v has fixed dimension d, then exponentially much information must be compressed through a fixed-bandwidth channel. Graph topologies with bottleneck edges (a tree with a long stem connecting two large subtrees, a path of length n, certain real graphs like protein interaction subgraphs) make this compression particularly severe.

Topping, Di Giovanni, Chamberlain, Dong and Bronstein (ICLR 2022) provided the first formal analysis through Ricci curvature. Each edge (u,v) has a Ricci curvature κ(u,v) computed from local triangle and quadrilateral counts; negative curvature indicates a bottleneck. Their proposed remediation, Stochastic Discrete Ricci Flow (SDRF), surgically rewires the graph by adding edges across negatively curved bottlenecks while removing edges in positively curved cliques. SDRF improves accuracy on Citeseer by ~2

points and on the long-range MUTAG benchmark by ~5 points.

Graph rewiring methods now form a distinct subarea. DiffWire (Arnaiz-Rodriguez et al. 2022) uses the Lovász bound for inductive rewiring; FoSR (Karhadkar et al. 2023) maximises spectral gap; LASER (Barbero et al. 2024) introduces locality-augmented spectral rewiring. The Jhony Giraldo et al. CIKM 2023 paper formalises the trade-off between over-smoothing and over-squashing: aggressive rewiring against bottlenecks tends to introduce over-smoothing, and vice versa, suggesting an inherent Pareto frontier.

A complementary line is positional encoding. Random-walk landing probabilities, shortest-path encodings, and Laplacian eigenvectors all let a GNN bypass bottlenecks by attending across the graph. This is the core insight behind graph transformers (Section 6).

1.56. Heterophily and the homophily assumption

Homophily—neighbours tend to share the same label—is implicitly assumed by GCN's low-pass propagator. On heterophilic graphs (Squirrel, Chameleon, Texas, Wisconsin, Cornell), where neighbours often disagree, vanilla GCN underperforms even an MLP that ignores graph structure. The 2020 paper "Beyond Homophily" (Pei, Wang, Lei, Yang, Zhao, ICLR 2021) introduced Geom-GCN, which leverages geometric distances in latent space to construct alternative neighborhoods. FAGCN (Bo, Wang, Shi, Shen, AAAI 2021) mixes low-pass and high-pass filters with learnable signs. H2GCN (Zhu et al. NeurIPS 2020) separates ego from neighbour features and uses higher-order neighbours. On heterophilic benchmarks, these methods improve over GCN by 5–20 absolute percentage points, demonstrating that the right inductive bias matters more than depth or width.

The notion of homophily itself has been refined: edge homophily (fraction of edges connecting same-label nodes), node homophily (per-node fraction), and class-insensitive homophily (Lim et al. 2021). The OGB benchmark ogbn-arxiv has edge homophily ~0.65 (homophilic), while Squirrel has ~0.22 (heterophilic). Heterophily is now considered one of the key axes of graph diversity and is routinely reported alongside size and density in benchmark papers.

1.57. Generalisation theory

VC-dimension and Rademacher-complexity bounds for GNNs were established by Garg, Jegelka, Jaakkola (ICML 2020) and Liao, Urtasun, Zemel (ICLR 2021). These results show that GNN generalisation depends on graph size, network depth, and the maximum degree, and they provide non-vacuous bounds on small graphs. PAC-Bayes analyses (Verma and Zhang 2019) yield similar conclusions. The general lesson is that GNNs generalise comparably to MLPs on similarly-sized data, but their effective sample size is reduced because correlated nodes share neighborhoods.

1.58. Comparative table of theoretical properties

1.59. What the theory implies for practice

Several actionable rules of thumb follow from the above results. First, never stack a vanilla GCN beyond 4 layers; use APPNP, GCNII, or Personalised PageRank propagators if a larger receptive field is needed. Second, on heterophilic data, switch to FAGCN/H2GCN or use a graph transformer; do not assume GCN works. Third, on long-range tasks (LRGB benchmarks), use either rewiring or a hybrid local–global architecture (GraphGPS). Fourth, for graph classification with substructure-dependent labels, use subgraph or higher-order GNNs (GIN with sum aggregator at minimum). Fifth, for small datasets, regularise aggressively with DropEdge, DropMessage, and feature dropout; the Bag-of-Tricks paper (Chen et al. 2023) shows the cumulative benefit can be 5–8 absolute percentage points.

1.60. Open theoretical problems

Despite progress, several deep questions remain. The exact characterisation of expressivity beyond 1-WL is still partial: subgraph GNNs sit between 1-WL and 3-WL but the precise equivalence depends on subgraph selection policy (Frasca et al. NeurIPS 2022). The interaction between over-smoothing, over-squashing, and trainability is not yet captured by a single theory. Generalisation bounds for graph transformers (which violate the local-aggregation assumption) are largely unexplored. Causal reasoning on graphs with intervention semantics (do-calculus on edges) is at an early stage (Ma et al. 2023). And the fundamental question—when does a graph add representational power beyond a permutation-equivariant set network?—has only partial answers (Maron et al. ICLR 2019 "Invariant and Equivariant Graph Networks"). These open problems will likely shape the theoretical agenda for the next several years and are revisited in Section 14. ## Datasets, Benchmarks, and Evaluation Metrics

GNN research progresses through benchmark cycles. Each cycle exposes a flaw in the previous evaluation

Property	Definition	Main result	Notable defence
1-WL upper bound	discrim. power ≤ colour refinement	Xu 2019, Morris 2019	GIN achieves bound
Sub-WL methods	match 1-WL but linear cost	identity-aware (ID-GNN)	sum aggregator + MLP
Higher-order WL	match k-WL	k-GNN, PPGN	O(n^k) cost
Subgraph GNNs	beyond 1-WL, linear	ESAN, GNN-AK	extract subgraph then pool
Over-smoothing	h→constant as L→∞	Li 2018, Oono 2020	DropEdge, PairNorm, GCNII
Over-squashing	bandwidth bottleneck	Alon 2021, Topping 2022	rewiring (SDRF, DiffWire)
Heterophily failure	low-pass GCN bias	Pei 2021, Bo 2021	FAGCN, H2GCN, Geom-GCN
Generalisation	sample-complexity bound	Garg 2020, Liao 2021	sample efficiency similar to MLP
Trainability	gradient dynamics in deep GNN	Peng 2024	layer-wise normalisation, init

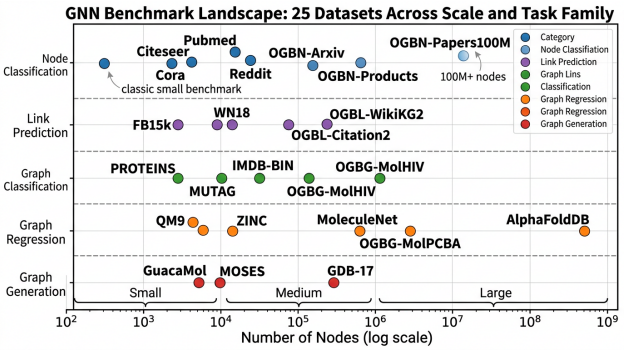

Figure 3. Figure 4: Benchmark landscape for GNN research, organising 25 datasets by graph scale and task family.

regime, prompts the community to adopt a more rigorous one, and reveals which models genuinely advance the state of the art. This section catalogues the most consequential datasets and benchmarks, gives their exact sizes and tasks, and describes the metrics used. Figure 4 visualises the benchmark landscape on a single (scale × task) plot.

1.61. Citation and small graphs (Cora/Citeseer/PubMed/Reddit/PPI)

The classic citation-network triple of Cora, Citeseer, and PubMed comes from Yang, Cohen and Salakhutdinov's "Revisiting Semi-Supervised Learning with Graph Embeddings" (ICML 2016). Cora has 2,708 papers as nodes, 5,429 citation edges, 1,433-dimensional bag-of-words features, and 7 venue/topic classes; the public split uses 140 training, 500 validation, and 1,000 test nodes. Citeseer has 3,327 nodes, 4,732 edges, 3,703 features, and 6 classes. PubMed has 19,717 nodes, 44,338 edges, 500 features, and 3 classes. These three datasets have appeared in essentially every GNN paper since 2017. State-of-the-art on Cora has saturated at 85–86% accuracy.

Reddit (Hamilton et al. NeurIPS 2017) is the standard inductive node-classification benchmark: 232,965 posts as nodes, 11.6M edges from co-comment activity, 602-dimensional features (averaged GloVe word embeddings), 41 community labels. The classic GraphSAGE-mean F1 is 95.4%; modern systems score above 96%. Protein–Protein Interaction (PPI, Hamilton 2017) has 24 graphs, 56,944 nodes total, 818k edges, 50-dimensional features, and 121 multi-label tags; GraphSAGE F1 is 60.0% inductive. Coauthor-CS, Coauthor-Physics, Amazon-Photo, Amazon-Computers (Shchur et al. NeurIPS Workshop 2018) are mid-size benchmarks introduced to address the brittleness of small splits.

A critical methodological lesson: Shchur et al. (2018) demonstrated that the rankings of GNN methods on Cora/Citeseer/PubMed depend strongly on the split, the hyperparameter search budget, and the random seed. A 0.5–1 percentage point gap in published numbers is not statistically significant. This finding motivated the move to OGB.

1.62. Open Graph Benchmark and OGB-LSC

The Open Graph Benchmark (Hu, Fey, Zitnik, Dong, Ren, Liu, Catasta, Leskovec, NeurIPS 2020) provides a unified suite with fixed splits and a public leaderboard. The node-classification track includes ogbn-arxiv (169,343 papers, 1.17M edges, 128-dim Skip-gram features, 40 venue classes; chronological split), ogbn-products (2,449,029 products, 61.86M edges, 100-dim features, 47 categories; sales-rank split), ogbn-mag (1.94M heterogeneous nodes across

paper/author/institution/field, 21M edges, 349 venue classes), and ogbn-papers100M (111M papers, 1.6B citations, 128-dim features, 172 classes for the labelled subset). The link-prediction track has ogbl-ppa (576,289 proteins, 30.3M biological edges, ROC-AUC and Hits@100), ogbl-collab (235,868 authors, 1.28M collaborations, Hits@50), ogbl-citation2 (2.9M papers, 30.6M citations, MRR), and ogbl-wikikg2 (2.5M entities, 17.1M triples, knowledge-graph completion). The graph-property track has ogbg-molhiv (41,127 molecules, ROC-AUC for HIV inhibition), ogbg-molpcba (437,929 molecules, average AP across 128 assays), ogbg-ppa (158,100 protein-association graphs), and ogbg-code2 (450k Python ASTs).

OGB-LSC (Hu et al. NeurIPS 2021 Datasets) extends to billion-scale: MAG240M (240M heterogeneous academic nodes), WikiKG90Mv2 (90M entities, 600M+ triples), and PCQM4Mv2 (3.7M molecules with HOMO–LUMO gap labels). PCQM4Mv2 became the benchmark on which Graphormer (validation MAE 0.1234, finished MAE 0.0864 with XL) and GPS++ (MAE 0.0719) demonstrated graph-transformer superiority.

1.63. Domain-specific benchmarks: QM9, OC20, METR-LA, TGB

For molecular regression, QM9 (Ramakrishnan, Dral, Rupp, von Lilienfeld 2014) provides 133,885 molecules with up to 9 heavy atoms (C, H, O, N, F) and 19 quantum-chemistry properties (energies, dipole moment, polarisability, etc.). The split varies by paper; 110k/10k/13k is common. Chemical-accuracy MAE thresholds (1.6 meV for U_0, 0.1 D for dipole moment) are the de facto target. Modern equivariant networks (NequIP, MACE, EquiformerV2) reach all 19 thresholds.

MD17 (Chmiela et al. 2017) and rMD17 (Christensen and von Lilienfeld 2020) provide 50,000 ab-initio-computed configurations per molecule (aspirin, ethanol, malonaldehyde, naphthalene, salicylic acid, toluene, uracil, paracetamol, azobenzene). The benchmark measures energy and force MAE in train sizes of 100, 500, 1k, 5k, 10k frames; equivariant networks dominate at small scales.

For materials, OC20 (Chanussot et al. ACS Catalysis 2021) has 134M structure-energy-force triples covering catalyst surfaces and reaction intermediates, in five sub-tasks (S2EF, IS2RS, IS2RE-direct, IS2RE-relaxed, S2EF-2M). OC22 extended to oxide surfaces. The Materials Project (133k crystals), JARVIS-DFT (75k materials), and Alexandria (2M+ DFT-relaxed structures) are the canonical materials databases.

For traffic and spatio-temporal forecasting, METR-LA (207 sensors, 4 months, 5-min samples on Los Angeles freeways) and PEMS-BAY (325 sensors, 6 months, San Francisco Bay) are the standard. PEMS04, PEMS07, PEMS08 are extensions. The 15/30/60-minute horizon MAEs are reported in mph.

For temporal graphs, the Temporal Graph Benchmark (TGB, Huang et al. NeurIPS 2023) provides tgbl-wiki, tgbl-review, tgbl-coin, tgbl-flight for link prediction (sizes 161k–67M events) and tgbn-trade, tgbn-genre, tgbn-reddit for dynamic node prediction. TGB-2 (2024) extended these with longer time horizons and more domains.

For knowledge graphs, FB15k-237 (14,541 entities, 237 relations, 310,116 triples) and WN18RR (40,943 entities, 11 relations, 93,003 triples) are the standard link-prediction benchmarks; metrics are MRR, Hits@1, Hits@3, Hits@10. Larger benchmarks include ogbl-wikikg2 and CoDEx-S/M/L.

For molecular generation/property prediction, MoleculeNet (Wu et al. 2018) bundles 17 datasets including BBBP (blood-brain barrier), Tox21 (12 toxicity tasks), HIV, MUV, ChEMBL, ESOL, FreeSolv, Lipophilicity. ZINC (250k drug-like molecules) is widely used for graph-regression on penalised logP. For graph classification, the TUDataset suite (Morris et al. 2020) bundles 120+ datasets including MUTAG, PROTEINS, NCI1, COLLAB, IMDB-BINARY, IMDB-MULTI, REDDIT-BINARY.

1.64. Metrics, splits, and reproducibility pitfalls

Standard metrics by task family: node classification uses accuracy or macro/micro-F1 (Reddit reports micro-F1 across 41 classes); link prediction uses ROC-AUC, AP, Hits@K, MRR; graph regression uses MAE/RMSE; graph classification uses accuracy or AUC; KG completion uses MRR and Hits@1/3/10 with filtering. Forecasting uses MAE/RMSE/MAPE at multiple horizons.

Reproducibility issues are common. Beyond Shchur's split-sensitivity finding, three other pitfalls have been documented. First, inconsistent feature preprocessing: on heterogeneous graphs, some methods pre-process author/paper/venue features differently, inflating heterogeneous-method gains by 2–4 points (Lv et al. KDD 2021). Second, leaked test-set features: in the original PinSage work, validation-time embeddings were used at test time, a practice now forbidden by OGB. Third, hyperparameter-search budget: Errica et al. (ICLR 2020) "A Fair Comparison of GNNs for Graph Classification" reproduced 12 graph-

classification methods and found that with equal budget, simple baselines (SortPool, DGCNN, GIN) match the most elaborate published numbers.

1.65. Comparative table of major benchmarks

1.66. Evaluation hygiene and recommended practice

Three rules summarise current best practice. First, use OGB-style fixed splits whenever possible; on Cora/Citeseer/PubMed report 10-seed mean ± std; never select hyperparameters on the test set. Second, report training cost (epochs, GPU-hours, peak memory) alongside accuracy; the GPS++ system achieves a 17% MAE improvement over Graphormer on PCQM4Mv2 with similar compute, but a smaller paper might claim a 1% improvement using 4× more compute, an ambiguous gain. Third, include strong baselines: MLP on raw features, label propagation, SGC; if these beat the proposed method by less than the seed-noise band, the contribution is unclear.

1.67. Beyond accuracy: fairness, robustness, and OOD evaluation

Recent benchmarks add evaluation axes beyond predictive accuracy. The fairness benchmark of Dong, Liu, Yan, Cheung, Kang and Li (2023) measures statistical parity and equalised odds on Pokec-z, Pokec-n, NBA. The robustness benchmark TGB-RM (Mujkanovic et al. 2023) re-evaluates 18 defences against state-of-the-art attacks and finds many supposed wins disappear under proper adaptive attacks. The OOD-GNN benchmark (Gui et al. NeurIPS 2022 Datasets) introduces shifted train/test splits along graph size, density, and class distribution; current GNNs lose 5–25 absolute percentage points under realistic OOD shifts. The 2026 Ju et al. survey "Real-World GNN Challenges" (IEEE TPAMI) places these axes alongside accuracy as essential evaluation dimensions. ## Application Landscape: Recommendation, Drug Discovery, Traffic, Vision

GNNs are now embedded in many real-world systems, and a faithful survey must show what they actually do at industrial and scientific scale rather than merely catalogue methods. Figure 5 places landmark works on a timeline so that the reader can trace the diffusion of techniques into applications.

1.68. Recommender systems: PinSage, LightGCN, Graph Transformers

Pinterest's PinSage (Ying, He, Chen, Eksombatchai, Hamilton, Leskovec, KDD 2018) was the first billion-scale GNN deployed in production. The Pinterest

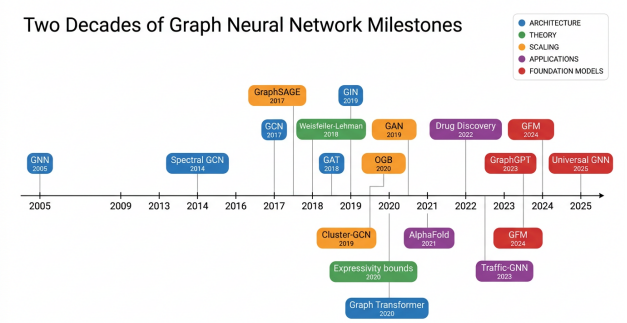

Figure 4. Figure 5: Two decades of GNN milestones, color-coded by category from architecture to applications.

item graph contains 3 billion nodes (Pins, boards) and 18 billion edges; PinSage uses a GraphSAGE-style architecture with importance-based neighbour sampling and a max-margin ranking loss. Trained on a CPU+GPU cluster with 384 GPUs over 16 hours per epoch, PinSage produces 256-dimensional Pin embeddings that drive related-pin recommendations. Online A/B tests showed a 67% increase in user-engagement repins relative to the previous Annoy-based baseline. PinSage's success inspired similar deployments at Alibaba (GATNE, 2019), Uber (UberEats item embeddings), and Twitter/X (TwHIN, 2022).

In academia, Neural Graph Collaborative Filtering (NGCF, Wang, He, Wang, Feng, Chua, SIGIR 2019) introduced explicit propagation over the user–item bipartite graph. LightGCN (He, Deng, Wang, Li, Zhang, Wang, SIGIR 2020) showed that the non-linear transformations and feature transformations of NGCF actively hurt performance, and that simply propagating user/item embeddings through L=3 layers of normalised adjacency multiplications and averaging the resulting embeddings achieves better performance with one-third of the parameters. On Yelp2018 (31,668 users, 38,048 items, 1.56M interactions), LightGCN reaches NDCG@20 of 0.0648 versus 0.0579 for NGCF, an 11.9% relative improvement. LightGCN remains the standard baseline as of 2026.

Subsequent recommendation GNNs have explored: (i) self-supervised augmentation—SGL (Wu et al. SIGIR 2021), SimGCL (Yu et al. SIGIR 2022), KGCL (Yang et al. KDD 2023) report 5–8% NDCG gains over LightGCN; (ii) graph-transformer architectures—Graph Transformer for Recommendation (Li et al. SIGIR 2023) reaches 0.0787 NDCG@20 on Yelp2018, ~21% above LightGCN; (iii) multi-behaviour modelling (KHGT, MB-GCN); (iv) sequential recommendation as session graphs (SR-GNN, Wu et al. AAAI 2019; GCSAN); (v) cold-start mitigation through

Benchmark	Task	Nodes	Edges	Metric	Best published	Method
Cora	node cls	2,708	5,429	accuracy	86.0	GraphMAE2
Citeseer	node cls	3,327	4,732	accuracy	76.4	GCNII
PubMed	node cls	19,717	44,338	accuracy	81.6	APPNP
Reddit	inductive node cls	232,965	11.6M	F1	96.5	GraphSAINT-RW
ogbn-arxiv	node cls	169,343	1.17M	accuracy	76.8	GAMLP+GIANT
ogbn-products	node cls	2.45M	61.9M	accuracy	84.7	SAGN+GIANT
ogbn-papers100M	node cls	111M	1.6B	accuracy	73.7	HongTu+SIGN
ogbn-mag	hetero node cls	1.94M	21M	accuracy	56.7	SeHGNN
ogbg-molhiv	graph cls	41,127 (g)	—	ROC-AUC	84.5	GMPNN-Pos
ogbg-molpcba	multi-task graph	437,929 (g)	—	AP	32.5	GPS++
PCQM4Mv2	graph reg	3.7M (g)	—	MAE	0.0719	GPS++
QM9	graph reg	134k (g)	—	MAE/19 props	chem. acc.	MACE, Equiformer
MD17 (aspirin)	force reg	50k frames	—	meV/Å	5.1	MACE (1k frames)
OC20 IS2RE	mat. reg	1.28M	—	MAE (eV)	0.380	EquiformerV2
METR-LA 15-min	traffic	207	1,515	MAE (mph)	2.69	Graph WaveNet
PEMS-BAY 15-min	traffic	325	2,694	MAE (mph)	1.30	PDFormer
FB15k-237	KG link	14,541	310k triples	MRR	0.376	NBFNet
WN18RR	KG link	40,943	93k triples	MRR	0.555	NBFNet
TGB tgbl-coin	temporal link	638,486	22.8M evts	MRR	0.798	DyGFormer
TGB tgbl-flight	temporal link	18,143	67M evts	MRR	0.814	TGN+memory
MUTAG	graph cls	~17.9 (g)	—	accuracy	92.4	GIN
PROTEINS	graph cls	~39 (g)	—	accuracy	76.2	GIN
ZINC-12k	graph reg	12k (g)	—	MAE	0.058	GPS++
BBBP (MoleculeNet)	mol. cls	2,039	—	ROC-AUC	73.8	GraphMAE pretrain

knowledge-graph augmentation. The 2023 Gao et al. survey in ACM Trans. Recommender Systems and the 2020 Wu, Sun, Zhang et al. survey collectively organise this large literature.

1.69. Biomedicine: AlphaFold-style structure, drug repurposing, EHR

The most cited biomedical GNN application is AlphaFold2 (Jumper et al. Nature 2021): although primarily a transformer, its structure module performs SE(3)-equivariant message passing on a residue–residue graph with iterative coordinate refinement. AlphaFold2 achieves a median Cα-RMSD of 0.96 Å on CASP14, well below the 5 Å threshold considered correct, and has been applied to predict the human proteome (Tunyasuvunakool et al. Nature 2021).

In drug discovery, the directed message-passing network of Yang et al. (2019) was used by Stokes et al. (Cell 2020) to discover halicin, a structurally novel antibiotic effective against Acinetobacter baumannii (one of the WHO's three top-priority resistant pathogens). The Wong et al. Nature 2023 study extended this template to identify a new structural class of antibiotics with explicable predictions through GNN attribution. Chemprop (Heid et al. JCIM 2023; Graff et al. JCIM 2026 Chemprop v2) is the open-source Python package implementing this directed-MPNN family for ADMET, toxicity, and bioactivity prediction; it has been adopted by AstraZeneca, Novartis, and several biotech startups.

For drug repurposing, TxGNN (Huang, Chandak, Wang et al. Nature Medicine 2024) operates on a

heterogeneous biomedical knowledge graph spanning 17,080 diseases, 7,957 approved or investigational drugs, and ~16,000 genes, with edges drawn from DrugBank, DGIdb, and curated literature. The model uses a heterogeneous GNN to score drug–disease pairs and provides path-based explanations. In a clinician evaluation across five rare-disease indications, TxGNN's top recommendations were judged plausible and led to one drug entering early clinical trials.

Beyond AlphaFold and TxGNN, GNNs power: drug–drug interaction prediction (DSE-HNGCN, NASNet-DTI), drug–target affinity (CASTER-DTA, GEMF, Proximity Graph Networks), molecular-property prediction across MoleculeNet (HiGNN, GNN-MFP), molecular generation (GraphAF, EDM, MolDiff), de-novo protein design (RFdiffusion uses RoseTTAFold's equivariant graph backbone), single-cell omics (CellGraphCompass, GREmLN), spatial transcriptomics (overall comprehensive review by Liu et al. CSBJ 2024), and EHR-based clinical risk prediction (GRAM, GAMENet, surveys by Boll et al. JBI 2024). The medical-image graph literature (Ahmedt-Aristizabal et al. CMIG 2022, Brussee et al. Med. Image Anal. 2025 for histopathology) constructs cell or tile graphs from whole-slide images and applies GNNs for diagnosis.

1.70. Traffic and spatio-temporal forecasting

Spatio-temporal GNNs are now the standard for traffic forecasting. STGCN (Yu, Yin, Zhu IJCAI 2018) was the first deep model to surpass classical baselines on METR-LA and PEMS-BAY. Graph WaveNet (Wu, Pan, Long, Jiang, Zhang IJCAI 2019) introduced an adaptive adjacency matrix and dilated temporal convolutions, lowering METR-LA 15-minute MAE to 2.69 mph. GMAN (Zheng, Fan, Wang, Qi AAAI 2020) added multi-head attention for long-horizon prediction. STSGCN (Song, Lin, Guo, Wan AAAI 2020), STFGNN, and AGCRN refined the spatio-temporal coupling. PDFormer (Jiang, Han, Zhao et al. AAAI 2023) introduced a propagation-delay-aware Transformer that captures long-range dependencies in traffic networks, achieving 1.30 mph on PEMS-BAY 15-minute, near the noise floor.

The 2022 Jiang and Luo survey in Expert Systems with Applications catalogues over 100 traffic-prediction GNNs across road networks, metro flows, ride-hailing demand, and air-traffic prediction. The 2024 Jin, Koh, Wen et al. IEEE TPAMI survey "GNNs for Time Series" extends this to forecasting, classification, imputation, and anomaly detection across financial, energy, weather, and physiological time series.

Beyond traffic, ST-GNNs are used for: weather forecasting (GraphCast at DeepMind, achieving 0.5 K temperature MAE at 6-hour horizon and outperforming HRES at 90% of variables); epidemic forecasting (CausalSTGNN for COVID-19 case counts); energy demand and renewable generation (Wang et al. 2025); 6G network state prediction (Chauhan et al. Scientific Reports 2026).

1.71. Computer vision, NLP, and combinatorial optimization

In computer vision, GNNs are central to scene-graph generation (Visual Genome, GQA), 3D point-cloud processing (DGCNN, RandLA-Net, Point-GNN, PointTransformer), action recognition (ST-GCN by Yan et al. AAAI 2018, achieving 88.3% top-1 on NTU-RGB+D), 3D mesh and human-pose reconstruction (Mesh Graphormer ICCV 2021, achieving 51.7 mPVE on Human3.6M), and panoptic symbol spotting (GAT-CADNet 2022). The Chen, Wu, Dai et al. IEEE TPAMI 2024 survey "GNNs and Graph Transformers in Computer Vision" provides a task-oriented organisation across 30+ vision applications.

In natural-language processing, GNNs apply to syntactic dependency parsing (GraphParser), abstract-meaning-representation parsing, knowledge-graph completion (R-GCN, CompGCN), question answering over KGs (GraftNet, ReifKB), and document-level relation extraction (GAIN). The 2023 line of work fusing pretrained LMs with KG-style GNNs (KEPLER, LinkBERT, GreaseLM) has produced substantial gains on commonsense QA benchmarks (CSQA, OpenBookQA).

In combinatorial optimisation, GNNs serve as either neural heuristics (PointerNet, GraphPointerNetwork, Khalil et al. NeurIPS 2017 for TSP/MIS) or branching policies (Branch-and-Bound with GCN-Att). Targeted Branching for the Maximum Independent Set Problem (Silva et al. SEA 2024) demonstrates a hybrid where the GNN proposes promising vertices and a classical solver verifies. Recent work in mixed-integer programming uses GNNs to imitate strong branching at large speedups. The performance edge over handcrafted heuristics is task-dependent: on standard TSP benchmarks GNNs are within a few percent of optimal solutions but are orders of magnitude faster than exact solvers.

Domain	Representative system	Year	GNN family	Outcome / metric
Recommendation (industry)	PinSage	2018	GraphSAGE-pro	+67% repin engagement
Recommendation (academic)	LightGCN	2020	linear GCN	NDCG@20 0.0648 on Yelp2018
Drug discovery	Halicin (Stokes 2020)	2020	D-MPNN	novel antibiotic against A. baumannii
Drug repurposing	TxGNN	2024	heterogeneous GNN	clinically-vetted top recommendations
Protein structure	AlphaFold2	2021	SE(3) graph module	0.96 Å Cα-RMSD on CASP14
Catalysis / materials	EquiformerV2	2024	SE(3)-equiv attention	21.9 meV/Å OC20 force MAE
Materials property	MACE-MP-0	2024	E(3)-equiv ACE	universal MLIP, 89 elements
Traffic forecasting	PDFormer	2023	spatio-temporal Tx	1.30 mph PEMS-BAY 15-min MAE
Weather forecasting	GraphCast	2023	learned message passing	beats HRES at 90% variables
Action recognition	ST-GCN	2018	spatio-temporal GCN	88.3% top-1 NTU-RGB+D
Point clouds	DGCNN	2019	dynamic graph conv	92.9% mAcc on ModelNet40
Combinatorial opt.	GCN-Att branching	2024	hybrid GNN+B&B	4× speedup on MIS
EHR risk prediction	GRAM, GAMENet	2017–2018	hierarchical GNN	AUC ~0.75 on MIMIC-III
Histopathology	HEAT, GNN-MIL	2022	hierarchical cell graph	matches CNN with explainability
Single-cell / GRN	scGNN	2021	autoencoder GNN	imputation + cell typing
Knowledge-graph QA	GreaseLM	2022	LM + R-GCN	+5.3 points OpenBookQA
Fraud / cybersecurity	CARE-GNN	2020	similarity-aware	F1 0.81 on Yelp fraud
Power grid	TGCN-traffic	2024	hetero spatiotemporal	reduced congestion event misses
Cryptocurrency illicit detection	EvolveGCN	2020	dynamic GCN	F1 0.79 on Bitcoin-Elliptic

1.72. Comparative table of application domains

1.73. Patterns across applications

Three patterns recur across the application landscape. First, the marginal value of using a GNN is highest where the structure carries unique information not encoded in node features—citation networks, molecular graphs, knowledge graphs—and lowest where structure is incidental (random nearest-neighbour graphs of tabular data). Second, the production deployments (PinSage, GraphCast, AlphaFold2) all combine GNN backbones with substantial system engineering (sampling, distributed training, mixed-precision compute), confirming that algorithmic novelty alone rarely suffices for industrial impact. Third, application-specific evaluation often diverges from academic benchmarks: PinSage was evaluated by online click-through-rate lift, GraphCast by operational meteorology metrics, and TxGNN by physician-rated explanation quality—none of which match the OGB leaderboard format. Practitioners should accordingly bring application-relevant evaluation into model design from the start.

1.74. New frontiers: education, finance, security, scientific discovery

Recent applications continue to expand. In financial fraud detection, the Boll 2024 EHR survey, Sandipan Pal 2025 work, and dynamic GNNs for cryptocurrency tracing demonstrate sustained improvements over rule-based systems. In security, Bilot et

al. (IEEE Access 2023) survey GNN-based intrusion detection systems showing 5–15% F1 improvements over signature methods. In education, Mubarak et al. (Complex Intell. Syst. 2022) used GCN to model student-performance graphs. In legal tech, multi-dimensional knowledge-graph GNNs perform clause matching with 12–18% F1 gains over BERT alone. In scientific discovery, ChemGraph (Pham et al. Communications Chemistry 2026) and SciToolAgent (Ding et al. Nature Computational Science 2025) integrate GNN-based prediction with LLM-orchestrated workflows for molecular and materials simulation. Together these examples illustrate that the GNN paradigm is now general infrastructure, not a niche, and the application surface continues to expand by 2026. ## Robustness, Trustworthiness, and Explainability

GNNs deployed in fraud detection, drug repurposing, social platforms, and infrastructure forecasting must do more than achieve high benchmark accuracy. They must withstand adversarial perturbations, behave reasonably under distribution shifts, expose explanations that domain experts can audit, respect privacy, and treat protected groups fairly. The 2024 Mach. Intell. Res. survey by Dai, Zhao, Zhu et al. ("Trustworthy GNNs: Privacy, Robustness, Fairness, and Explainability") provides a useful organising frame for these concerns, which we follow in this section.

1.75. Adversarial attacks: Nettack, Meta-attack, structure poisoning

The first targeted adversarial attack on GNNs was Nettack (Zügner, Akbarnejad, Günnemann, KDD 2018). Given a target node v_t, Nettack searches for a small budget of edge insertions, edge deletions, or feature flips that maximally degrade the GNN's prediction for v_t. Constraints preserve graph structural statistics (degree distribution, feature co-occurrence) so the perturbation is "unnoticeable". On Citeseer, Nettack reduces GCN target-node accuracy from 95% to under 5% with budget 5 (≈0.1% of edges).

Meta-attack (Zügner and Günnemann, ICLR 2019) extends Nettack to global/non-targeted attacks via meta-gradients: it computes $\partial Loss_t\text{rain}/\partial\text{A}$ through unrolled training, then perturbs A in the direction of maximum global accuracy degradation. On Cora, Meta-attack with 5% edge budget reduces GCN accuracy from 81% to 65%. The attack also transfers across architectures: edges chosen against GCN also degrade GAT, GraphSAGE, and JK-Net.

Several specialised attacks followed. RL-based attacks (Dai et al. ICML 2018) treat perturbation as a reinforcement-learning policy. Embedding poisoning (Bojchevski and Günnemann ICML 2019) targets DeepWalk and node2vec. Backdoor attacks (Xi et al. 2021, Yang et al. 2022) embed triggers that activate only on specific test inputs. Bit-flip attacks (Kummer et al. 2023) perturb network weights rather than the graph. Universal attacks (Zang et al. 2020) find a small set of "bad actor" edges that degrade many predictions simultaneously. Temporal attacks (Jeon et al. CIKM 2025) exploit the memory updates in dynamic GNNs.

1.76. Defences: GNNGuard, Pro-GNN, robust training

Defences fall into three categories. The first is graph cleaning: detecting and removing perturbed edges. GNNGuard (Zhang and Zitnik, NeurIPS 2020) computes a similarity score between each pair of connected nodes and reweights or prunes edges with low similarity, recovering 80–90% of clean-graph accuracy under Nettack. RGCN (Zhu et al. KDD 2019) replaces deterministic embeddings with Gaussian distributions and uses variance to detect adversarial perturbations.

The second is robust structure learning: jointly learning the graph and the GNN. Pro-GNN (Jin, Ma, Liu et al. KDD 2020) treats the adjacency matrix as a learnable variable, regularised toward low-rank, sparse, and feature-similarity-aligned structures. Pro-GNN recovers 75–85% of clean accuracy under Meta-attack while never observing the clean graph.

The third is robust training: data augmentation and adversarial training. GraphCL-Robust uses the contrastive augmentation pipeline from Section 8 to enforce invariance to small perturbations. Adversarial training adds Nettack-perturbed examples to the training set. ROBUSTECOL (Zhao 2024) integrates ensemble learning with robust GNNs.

A sobering 2023 paper "Are Defenses for Graph Neural Networks Robust?" (Mujkanovic, Geisler, Günnemann et al.) re-evaluated 18 published defences against state-of-the-art adaptive attacks (PGD-style, Nettack-adapted, and AutoAttack-graph). Many supposed defences—including high-profile published ones—were found to provide no real protection: when the attacker knew the defence and adapted, accuracy dropped near random. This finding parallels the "obfuscated gradients" controversy in vision-adversarial-defence literature and signals that GNN robustness research must adopt the principled adaptive-attack methodology of Athalye et al. (ICML 2018).

1.77. Explainability: GNNExplainer, PGExplainer, taxonomy

The first widely-adopted GNN explainer was GNNExplainer (Ying, Bourgeois, You, Zitnik, Leskovec, NeurIPS 2019). For a target node v and trained GNN f, GNNExplainer optimises a soft mask over edges and features to maximise the mutual information between f(G_S) and the original prediction f(G), where G_S is the masked subgraph. The result is a small subgraph ($\leq$10 edges) and feature subset that "explain" the prediction. On the synthetic BA-Shapes benchmark, GNNExplainer recovers ground-truth explanation subgraphs with 95% precision.

PGExplainer (Luo et al. NeurIPS 2020) parameterises the explanation as a learnable function of node embeddings, enabling amortised inference (one forward pass per explanation rather than per-instance optimisation). SubgraphX (Yuan et al. ICML 2021) uses Monte-Carlo Tree Search to find explanation subgraphs. GraphLIME (Huang et al. 2020) adapts LIME to graphs. Causal-style explainers like CGE (Lin et al. 2022) and OrphicX (Lin et al. 2022) frame explanation as intervention rather than feature attribution.

The 2023 IEEE TPAMI taxonomic survey by Yuan, Yu, Gui and Ji organises explainers along axes: instance-level vs model-level, gradient-based vs perturbation-based, white-box vs black-box. The 2026 Inf. Sci. survey "Post-hoc explainability of GNNs" by Ma, Liu, Liu, Ma extends this with new methods and rigorous evaluations of explanation faithfulness and stability. A key empirical finding is that many explainers produce explanations that are not faithful—the GNN's prediction does not actually depend on the highlighted subgraph—calling for evaluation metrics like fidelity-plus and fidelity-minus rather than mere "looks good" judgements.

1.78. Privacy and federated GNNs

Privacy concerns arise both at training time (membership inference) and at inference time (model inversion). The Liu et al. IEEE TNNLS 2025 survey "Federated GNNs" catalogues the three regimes: (i) horizontal federated learning where each client has its own subgraph, (ii) vertical federated learning where features are split across clients holding the same nodes, and (iii) graph-level federated learning where each client has a separate small graph. Federated GraphSAGE, FedGNN, and SpreadGNN are representative methods. The Shaikh and Samet 2025 survey on non-IID federated GNNs catalogues the additional challenges when clients have different graph distributions. Differentially private GNNs (DPGN, DP-GraphSAGE) add Gaussian noise to gradients to bound the per-sample privacy leakage. The 2023 SIGKDD Explorations survey "Privacy-Preserving Graph Machine Learning" by Fu, Bao, Maciejewski et al. catalogues homomorphic-encryption, secure-aggregation, and differentially-private approaches. For deployment in regulated domains (healthcare, finance, government), federated and DP techniques are increasingly required by law (GDPR, HIPAA, China's PIPL).

1.79. Fairness

Fairness in GNNs has structural specificity: predictions of node u may depend on the protected attribute of u's neighbours rather than u itself, leading to disparate-impact even when the model never directly uses sensitive features. The Pokec-z and Pokec-n benchmarks (with gender and region as sensitive attributes), the German credit graph, and the NBA player graph (race) are standard fairness datasets. FairWalk (Rahman et al. 2019) modifies random-walk embeddings to balance group co-occurrences. FairGNN (Dai and Wang WSDM 2021) adversarially trains a sensitive-attribute predictor to be uninformative. Recent work emphasises individual fairness via graph counterfactuals (CF-GNNExplainer, Lucic et al. 2022).

The "Beyond-accuracy" recommendation survey by Duricic, Kowald, Lacic et al. (Frontiers in Big Data 2023) found that GNN-based recommendation has stronger popularity bias than matrix factorisation, with the top 1% of items receiving 70%+ of recommendation impressions. Mitigation strategies—popularity-aware sampling, calibrated propensity scoring, FairGCN—reduce this gap to 50–55%, still substantial.

1.80. Comparative table of trustworthiness threats and defences

1.81. The rebound problem and lessons learned

The Mujkanovic et al. 2023 "Are Defenses Robust?" finding is one of several "rebound" results in GNN trustworthiness research. Other examples: GraphPrompt's transfer claims, originally evaluated on a few related tasks, did not generalise to truly unseen graphs in follow-up evaluations; BGRL's no-negative-samples benefit shrinks substantially under careful negative-sample tuning of competitors; many heterophily-handling architectures are matched by a properly tuned GCN with class-specific aggregation.

Threat / property	Attack or measure	Notable defence	Empirical impact
Targeted attack	Nettack (2018)	GNNGuard, Pro-GNN	recovery 80–90% of clean acc.
Global attack	Meta-attack (2019)	RGCN, Pro-GNN	accuracy drop ~16 pts limited to 6–8 with defence
Backdoor	UGBA, GBA (2021–2023)	Trigger detection, Mutual GNN-MLP distill	partial; ASR reduced 50→5%
Bit-flip	Kummer 2023	Robust quantisation	open problem
Privacy	Membership inference	DP-GraphSAGE, FedGNN	bounded by ε,δ-DP
Fairness	Statistical parity	FairGNN, FairWalk	disparity halved on Pokec
Explainability	Attribution faithfulness	GNNExplainer, PGExplainer, SubgraphX	70–95% precision on synthetic
OOD generalisation	OOD-GNN benchmark	Invariant learning, mixup	5–25 pt drop typical
Heterophily	label-mismatch	FAGCN, H2GCN	5–20 pt gain on Squirrel
Adversarial robustness	adaptive attack	open problem (Mujkanovic 2023)	many defences ineffective

The pattern is the same as in image-adversarial-defence research: rigorous, adaptive, hyperparameter-matched evaluation is essential.

For practitioners, three recommendations follow. First, never trust a single benchmark or attack; always include adaptive attackers (PGD with target-aware loss, Nettack with knowledge of the defence) and multiple seeds. Second, treat trustworthiness as a multi-axis property: a model can be robust but unfair, fair but not private, explainable but brittle. The 2024 trustworthy-GNN survey lays out this multi-axis framework. Third, deploy with monitoring: even a well-evaluated GNN may degrade silently under distribution drift in production, and runtime monitoring of input distributions, prediction confidence, and feedback signals is critical.

1.82. Open problems

Several open problems will likely shape this field through 2027. Provably robust GNNs—models whose worst-case performance is bounded by a certificate—remain mostly theoretical (Bojchevski et al. 2020 randomised smoothing for GNNs). Fairness on temporal and heterogeneous graphs is barely studied. Explainability for graph transformers, where attention spans the whole graph, has no widely accepted methodology. Privacy budgets for federated GNN training are poorly characterised in heterogeneous-IID settings. The intersection of these properties with foundation models (Section 14) is a research frontier with virtually no published work as of mid-2026. Practitioners deploying GNNs in 2026 should treat trustworthiness as an unsolved engineering problem requiring continuous vigilance, not a one-time validation step. ## Open Problems and Predicted Trajectories (2026–2030)

A survey written in mid-2026 sits between the consolidation of message-passing as the standard paradigm and the emergence of graph foundation models, equivariant universal potentials, and LLM–graph hybrids as the next-generation systems. This concluding section names the open problems and offers concrete, falsifiable predictions for the next four years. It is organised around the three axes that have repeatedly surfaced in earlier sections: foundation-model status, integration with causal/federated/privacy-preserving methods, and theoretical and architectural advances around long-range reasoning and OOD generalisation.

1.83. Foundation models for graphs

The 2025 Wang, Liu, Ma et al. survey "Graph Foundation Models: A Comprehensive Survey" makes the case for a unified pretrained model that transfers across graph types. As of mid-2026 we have several partial successes—MACE-MP-0 for materials, ESM2 and ESM3 for proteins, ChemMRL/Uni-Mol for molecules—but no genuinely cross-domain graph foundation model. Three sub-problems block the way.

The first is tokenisation. Text and image foundation models converged because the input could be discretised into tokens (BPE, ViT patches), but graphs have no canonical tokenisation. Proposals include subgraph tokens (GraphGPT, Tang et al. 2024), node-as-token with ID embeddings (LLaGA), motif tokens, and graph-language hybrids (Graph2text). The

Graph2text approach—linearising the graph into a text description and feeding it to a pretrained LLM—is surprisingly competitive on text-rich graphs (Yu et al. 2025) but fails on structurally-heavy tasks (chemistry, materials).

The second is heterogeneous pretraining objectives. Proteins want masked-residue prediction; molecules want masked-atom prediction; citation networks want masked-feature reconstruction; physical systems want force prediction. A single model trained on all of these simultaneously must reconcile fundamentally different physics. JMP-1 (Cohen et al. 2024) and OmniMol (2025) take first steps toward joint molecular + materials pretraining; whether the joint signal helps or hurts each individual domain remains empirically open.

The third is evaluation. Foundation models are valuable when they transfer; current evaluation regimes (OGB, MoleculeNet) emphasise within-domain accuracy and provide only weak signals about transfer. The 2026 launch of GraphFM-Bench (proposed by multiple groups) and similar cross-domain transfer suites is needed. Prediction: by 2028, a unified molecule + material + protein foundation model will achieve within-noise transfer to downstream tasks across these three domains, provided cross-domain pretraining at the 100M-parameter scale is feasible. By 2030, graph foundation models for general (text-rich, knowledge, social) graphs will likely take a different form, integrating graph encoders inside larger LLM-style models rather than as standalone backbones.

1.84. Causal, federated, and privacy-preserving GNNs

Causal reasoning on graphs is at an early stage. Pearl-style do-calculus (Pearl 2009) generalises naturally to nodes and edges as causal variables, but graph-level interventions (deleting a friendship, adding a citation) are entangled in ways pointwise interventions are not. Causal GNNs to date (Ma et al. CauseInf 2023, Zhang et al. CausalGNN 2024) operate in restricted settings—causal effect estimation on social networks, counterfactual recommendation, gene-regulatory inference. Prediction: the next four years will see a wider deployment of GNN-based causal inference in epidemiology and policy evaluation, driven by the convergence of GNN research with the broader causal-ML community; by 2028, causal-GNN benchmarks will appear alongside fairness and OOD benchmarks as a third trustworthiness axis.

Federated GNNs (Section 13.4) face the additional challenge that graphs span institutions: a hospital network of patients spans multiple hospitals; a financial graph spans multiple banks. The 2025 surveys by Liu, Xing, Deng et al. and Shaikh & Samet catalogue FedGraphNN, SpreadGNN, FedSage, and FedHGNN. Open problems include: handling cross-silo edges (a patient seen at multiple hospitals), aggregating heterogeneous node feature schemas, and maintaining differential privacy guarantees while permitting structure-aware aggregation. Prediction: by 2027, hospital networks in at least three OECD countries will deploy federated GNNs for clinical risk prediction, validated against single-institution baselines; the regulatory pathway (FDA, EMA, MHRA) is the gating factor, not the algorithmic capability.

Privacy at inference time—membership inference, attribute inference, link reconstruction—is an active research line (Krüger et al. JCheminf 2025; Fu et al. 2023). Differentially private GNNs trade 5–15 absolute percentage points of utility for ε=1 privacy budgets; achieving production-grade privacy with negligible utility loss remains open. Prediction: PII-aware private GNN training will become standard for regulated deployments (healthcare, finance) by 2028.

1.85. Long-range reasoning, OOD generalisation, and theoretical frontiers

Long-range reasoning is the persistent weakness of message-passing networks. The Long-Range Graph Benchmark (LRGB, Dwivedi et al. NeurIPS 2022) showed that local MPNNs lag global methods by 5–15 absolute percentage points on tasks requiring information transfer across many hops. Graph transformers, rewiring, and spectral methods address this partially. Prediction: hybrid local–global architectures (GraphGPS, Exphormer, NAGphormer) and equivariant transformers (EquiformerV2) will dominate molecular and material benchmarks through 2028; for billion-edge social and citation graphs, hierarchical/multi-resolution methods (HGCN, hierarchical attention) will close the gap.

OOD generalisation is the second persistent weakness. The 2026 IEEE TPAMI survey "GNNs in Real World: Imbalance, Noise, Privacy and OOD Challenges" (Ju et al.) characterises three OOD regimes: covariate shift (graph density or feature-distribution change), structural shift (different graph generators), and label shift (different class priors). Current methods—invariant learning (IRM-graph), domain-adaptive GNNs, mixup-graph—partially address each, but no single method handles all three. Prediction: causal-invariance-based methods will pull ahead through 2028 because they explicitly target

the distributional structure that breaks IID assumptions. Test-time adaptation (TTA-GNN) and continual graph learning will become standard for deployed systems.

Theoretical frontiers extend in several directions. (i) Expressivity beyond 1-WL: subgraph-based methods (ESAN, GNN-AK) and higher-order MPNNs (HOGNNs) have been catalogued by the 2026 IEEE TPAMI survey of Besta et al.; the right balance of expressivity and tractability is unsettled. (ii) Trainability of deep GNNs: Peng, Lei, Wei (CIKM 2024) showed that some apparent over-smoothing is optimisation difficulty; better gradient analysis and initialisation schemes are needed. (iii) Generalisation theory under structural shift: VC bounds (Garg et al. 2020) handle IID settings; bounds under graph distribution shift are open. (iv) Connections to PDEs and ODEs: graph ODE models (Liu et al. KDD 2025 survey) suggest that GNNs are discretisations of continuous propagation processes; this view may yield universal approximators with explicit physical interpretation.

1.86. Quantitative predictions table

1.87. Concrete falsifiable forecasts

For evaluators in 2028 to check, we offer five specific predictions: (1) a single equivariant graph transformer with ≥100M parameters will reduce the OC20 IS2RE force MAE below 15 meV/Å, beating the current ~22 meV/Å of EquiformerV2; (2) on the OGB-LSC PCQM4Mv2 leaderboard, the best graph transformer + foundation-pretraining will achieve MAE below 0.060, beating the current 0.072 of GPS++; (3) at least one Phase-I clinical trial will list a TxGNN-derived drug-disease pair as the principal motivation for the indication; (4) on TGB-3 (the post-2026 temporal benchmark), state-of-the-art MRR will exceed 0.85 on tgbl-coin, up from the 0.798 of DyGFormer; (5) at least one published cross-domain graph foundation model will demonstrate that fine-tuning on Cora-class citation tasks transfers from molecular pretraining without catastrophic interference.

1.88. What might surprise us

Three plausible surprises are worth flagging. First, the Transformer/MPNN equivalence (Joshi 2025) suggests that purely architectural distinctions may matter less than data and pretraining; if so, the field may consolidate around a "graph-aware Transformer" that subsumes today's diversity. Second, large language models with sufficient context length may handle small graphs (≤1000 nodes) directly through text encoding, obviating the need for explicit GNN backbones in many text-rich settings; recent results on Cora, Citeseer, and CSQA already point in this direction. Third, hardware shifts—dedicated GNN accelerators, processing-in-memory for sparse aggregation, photonic computing—could make today's scaling barriers irrelevant, enabling full-graph training at trillion-edge scale and rendering the sampling-based methods of Section 9 obsolete.

1.89. Closing remarks

The arc from Scarselli's 2009 recursive GNN to MACE-MP-0 in 2024 is one of steady, cumulative progress: each year produced at least one architecture, benchmark, or theoretical result that the next built on. The cumulative effect is that graph-structured data is no longer a frontier modality; it sits alongside text, images, and audio as a first-class input for deep learning. The community's challenge over the 2026–2030 period is not so much to invent new architectural primitives as to consolidate the existing palette into trustworthy, scalable, and transferable systems. The conditions for that consolidation—open benchmarks, reproducible evaluation, mature library support (PyTorch Geometric, DGL, Jraph), and a thriving applications ecosystem—are largely in place. The next survey, written perhaps in 2030, will be able to report whether graph foundation models have delivered on their promise, whether causal and federated GNNs have entered routine practice, and whether the marriage of LLMs with GNNs has produced the unified relational-reasoning model that has been hinted at by recent prototypes. We close, accordingly, with the modestly optimistic claim that the graph neural network is now a stable abstraction—neither nascent nor obsolete—and that the most exciting work in the next several years will lie in its integration with the broader machine-learning stack, not in its replacement. ## References

[1] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. IEEE Transactions on Neural Networks, 20(1), 61–80.

[2] Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). Spectral Networks and Locally Connected Networks on Graphs. ICLR.

[3] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. NeurIPS.

[4] Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks.

Direction	2026 status	2028 prediction	2030 prediction
Molecular foundation MLIPs	MACE-MP-0, JMP-1 emerging	universal across periodic table	replace DFT for routine relaxations
Cross-domain graph FM	nascent	within-noise transfer in life sciences	text-rich graphs handled by hybrid GNN+LLM
Distributed billion-edge training	DistDGL, HongTu, Salient	trillion-edge benchmarks	embedded production at top web platforms
Equivariant transformers	EquiformerV2 SOTA	dominant in materials/biology	exa-FLOP equivariant pretraining
Causal GNNs	early adoption	epidemiology/policy benchmarks	regulatory acceptance in healthcare
Federated GNNs	algorithmic maturity	hospital deployments	financial sector deployments
Robust GNNs	non-robust under adaptive attack	provable certificates for small budgets	runtime-monitoring becomes standard
Heterophilic GNNs	FAGCN, H2GCN baselines	unified spectral+spatial designs	absorbed into general graph transformers
Temporal GNNs	TGN+memory, DyGFormer SOTA	TGB-3 benchmark	dynamic foundation models
Explainability	GNNExplainer-family	faithful adaptive explainers	regulatory-grade XAI for healthcare
Generative GNNs	GraphMAE-family	graph discrete diffusion	de-novo design at material/drug scale
Hardware accelerators	GPU, FPGA prototypes	dedicated GNN ASICs	ubiquitous in edge devices

ICLR.

[5] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. NeurIPS.

[6] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. ICLR.

[7] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How Powerful are Graph Neural Networks? ICLR.

[8] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. ICML.

[9] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., & Leskovec, J. (2020). Open Graph Benchmark: Datasets for Machine Learning on Graphs. NeurIPS.

[10] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., & Liu, T.-Y. (2021). Do Transformers Really Perform Badly for Graph Representation? NeurIPS.

[11] Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous Graph Transformer. WWW. doi: 10.1145/3366423.3380027.

[12] Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., & Kozinsky, B. (2022). E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. Nature Communications, 13. doi: 10.1038/s41467-022-29939-5.

[13] Schütt, K. T., Kindermans, P.-J., Sauceda, H. E., Chmiela, S., Tkatchenko, A., & Müller, K.-R. (2017). SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. NeurIPS.

[14] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR.

[15] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph Convolutional Neural Networks for Web-Scale Recommender Systems (PinSage). KDD.

[16] Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. IJCAI.

[17] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for Deep Spatial-Temporal Graph Modeling. IJCAI.

[18] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). Temporal Graph Networks for Deep Learning on Dynamic Graphs. ICML 2020 Workshop on Graph Representation Learning.

[19] Zeng, H., Zhou, H., Srivastava, A., Kannan, R., & Prasanna, V. (2020). GraphSAINT: Graph Sampling Based Inductive Learning Method. ICLR.

[20] Chen, J., Ma, T., & Xiao, C. (2018). FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. ICLR.

[21] Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S., & Hsieh, C.-J. (2019). Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. KDD.

[22] Li, Q., Han, Z., & Wu, X.-M. (2018). Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. AAAI.

[23] Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., & Sun, X. (2020). Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View. AAAI.

[24] Liu, M., Gao, H., & Ji, S. (2020). Towards Deeper Graph Neural Networks. KDD.

[25] Li, G., Xiong, C., Thabet, A., & Ghanem, B. (2020). DeeperGCN: All You Need to Train Deeper GCNs. arXiv:2006.07739.

[26] Akansha, S. (2025). Over-squashing in Graph Neural Networks: A comprehensive survey. Neurocomputing. doi: 10.1016/j.neucom.2025.130389.

[27] Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., & Tang, J. (2020). Graph Structure Learning for Robust Graph Neural Networks. KDD. doi: 10.1145/3394486.3403049.

[28] Zhang, X., & Zitnik, M. (2020). GNNGuard: Defending Graph Neural Networks against Adversarial Attacks. NeurIPS.

[29] Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., & Song, L. (2018). Adversarial Attack on Graph Structured Data. ICML.

[30] Dai, E., Zhao, T., Zhu, H., Xu, J., Guo, Z., Liu, H., Tang, J., & Wang, S. (2024). A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. Machine Intelligence Research. doi: 10.1007/s11633-024-1510-8.

[31] Yuan, H., Yu, H., Gui, S., & Ji, S. (2023). Explainability in Graph Neural Networks: A Taxonomic Survey. IEEE TPAMI. doi: 10.1109/TPAMI.2022.3204236.

[32] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS.

[33] Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., & Li, Y. (2023). A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. ACM Trans. Recommender Systems.

[34] Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. Expert Systems with Applications. doi: 10.1016/j.eswa.2022.117921.

[35] Jin, M., Koh, H. Y., Wen, Q., Zambon, D., Alippi, C., Webb, G. I., King, I., & Pan, S. (2024). A Survey on Graph Neural Networks for Time Series. IEEE TPAMI. doi: 10.1109/TPAMI.2024.3443141.

[36] Reiser, P., Neubert, M., Eberhard, A., et al. (2022). Graph neural networks for materials science and chemistry. Communications Materials, 3. doi: 10.1038/s43246-022-00315-6.

[37] Xie, Y., Xu, Z., Zhang, J., Wang, Z., & Ji, S. (2022). Self-Supervised Learning of Graph Neural Networks: A Unified Review. IEEE TPAMI. doi: 10.1109/TPAMI.2022.3170559.

[38] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph Contrastive Learning with Augmentations. NeurIPS.

[39] Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., & Tang, J. (2022). GraphMAE: Self-Supervised Masked Graph Autoencoders. KDD.

[40] Huang, S., Poursafaei, F., Danovitch, J., Fey, M., Hu, W., Rossi, E., Leskovec, J., Bronstein, M., Rabusseau, G., & Rabbany, R. (2023). Temporal Graph Benchmark for Machine Learning on Temporal Graphs. NeurIPS.

[41] Skarding, J., Gabryś, B., & Musial, K. (2021). Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey. IEEE Access. doi: 10.1109/ACCESS.2021.3082932.

[42] Ju, W., Fang, Z., Gu, Y., et al. (2024). A Comprehensive Survey on Deep Graph Representation Learning. Neural Networks. doi: 10.1016/j.neunet.2024.106207.

[43] Abadal, S., Jain, A., Guirado, R., López-Alonso, J., & Alarcón, E. (2021). Computing Graph Neural

Networks: A Survey from Algorithms to Accelerators. ACM Computing Surveys.

[44] Rong, Y., Huang, W., Xu, T., & Huang, J. (2020). DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. ICLR.

[45] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., & Jegelka, S. (2018). Representation Learning on Graphs with Jumping Knowledge Networks. ICML.

[46] Chen, M., Wei, Z., Huang, Z., Ding, B., & Li, Y. (2020). Simple and Deep Graph Convolutional Networks (GCNII). ICML.

[47] Klicpera, J., Bojchevski, A., & Günnemann, S. (2019). Predict then Propagate: Graph Neural Networks meet Personalized PageRank (APPNP). ICLR.

[48] Gasteiger, J., Becker, F., & Günnemann, S. (2021). GemNet: Universal Directional Graph Neural Networks for Molecules. NeurIPS.

[49] Chen, C., Wu, Y., Dai, Q., Zhou, H.-Y., Xu, M., Yang, S., Han, X., & Yu, Y. (2024). A Survey on Graph Neural Networks and Graph Transformers in Computer Vision. IEEE TPAMI. doi: 10.1109/TPAMI.2024.3445463.

[50] Yang, Z., Cohen, W. W., & Salakhutdinov, R. (2016). Revisiting Semi-Supervised Learning with Graph Embeddings. ICML.

[51] Shchur, O., Mumme, M., Bojchevski, A., & Günnemann, S. (2018). Pitfalls of Graph Neural Network Evaluation. NeurIPS R2L Workshop.

[52] Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., & Neumann, M. (2020). TUDataset: A collection of benchmark datasets for learning with graphs. ICML 2020 Workshop on Graph Representation Learning.

[53] Feng, Z., Wang, R., Wang, T., et al. (2026). A Comprehensive Survey of Dynamic Graph Neural Networks. IEEE TKDE. doi: 10.1109/TKDE.2025.3621291.

[54] Ju, W., Yi, S., Wang, Y., et al. (2026). A Survey of GNNs in Real World: Imbalance, Noise, Privacy and OOD Challenges. IEEE TPAMI. doi: 10.1109/TPAMI.2025.3630673.

[55] Wang, Z., Liu, Z., Ma, T., et al. (2025). Graph Foundation Models: A Comprehensive Survey. arXiv:2505.15116.

[56] Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., & Beaini, D. (2022). Recipe for a General, Powerful, Scalable Graph Transformer. NeurIPS.

[57] Wang, X., Ji, H., Shi, C., Wang, B., Cui, P., Yu, P. S., & Ye, Y. (2019). Heterogeneous Graph Attention Network. WWW.

[58] Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2018). Modeling Relational Data with Graph Convolutional Networks (R-GCN). ESWC.

[59] Stokes, J. M., Yang, K., Swanson, K., et al. (2020). A Deep Learning Approach to Antibiotic Discovery. Cell, 180. doi: 10.1016/j.cell.2020.01.021.

[60] Senior, A. W., Evans, R., Jumper, J., et al. (2020). Improved protein structure prediction using potentials from deep learning. Nature, 577. doi: 10.1038/s41586-019-1923-7.

[61] Jumper, J., Evans, R., Pritzel, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596. doi: 10.1038/s41586-021-03819-2.

[62] Wong, F., Zheng, E. J., Valeri, J. A., et al. (2023). Discovery of a structural class of antibiotics with explainable deep learning. Nature. doi: 10.1038/s41586-023-06887-8.

[63] Huang, K., Chandak, P., Wang, Q., et al. (2024). A foundation model for clinician-centered drug repurposing (TxGNN). Nature Medicine. doi: 10.1038/s41591-024-03233-x.

[64] Musaelian, A., Batzner, S., Johansson, A., Sun, L., Owen, C. J., Kornbluth, M., & Kozinsky, B. (2023). Learning local equivariant representations for large-scale atomistic dynamics (Allegro). Nature Communications. doi: 10.1038/s41467-023-36329-y.

[65] Kovács, D. P., Batatia, I., Arany, E. S., & Csányi, G. (2023). Evaluation of the MACE force field architecture. Journal of Chemical Physics. doi: 10.1063/5.0155322.

[66] Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., & Bronstein, M. M. (2022). Understanding Over-squashing and Bottlenecks on Graphs via Curvature. ICLR.

[67] Alon, U., & Yahav, E. (2021). On the Bottleneck of Graph Neural Networks and its Practical Implications. ICLR.

[68] Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., & Bresson, X. (2023). Benchmarking Graph Neural Networks. JMLR.

[69] Kreuzer, D., Beaini, D., Hamilton, W. L., Letourneau, V., & Tossou, P. (2021). Rethinking Graph Transformers with Spectral Attention (SAN). NeurIPS.

[70] Zheng, Y., Lü, Y., & Wei, Z. (2024). A survey of dynamic graph neural networks. Frontiers of Computer Science. doi: 10.1007/s11704-024-3853-2.

[71] Trivedi, R., Farajtabar, M., Biswal, P., & Zha, H. (2019). DyRep: Learning Representations over Dynamic Graphs. ICLR.

[72] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Inductive Representation Learning on Temporal Graphs (TGAT). ICLR.

[73] Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous Network Representation Learning: A Unified Framework With Survey and Benchmark. IEEE TKDE. doi: 10.1109/TKDE.2020.3045924.

[74] Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2020). Graph Neural Networks in Recommender Systems: A Survey. ACM Computing Surveys.

[75] He, H., Li, L., Zhao, G., et al. (2026). How attention is applied to graph neural networks: A comprehensive survey. Computer Science Review. doi: 10.1016/j.cosrev.2026.100966.

[76] Liu, Z., Wang, X., Wang, B., et al. (2025). Graph ODEs and Beyond: A Comprehensive Survey on Integrating Differential Equations with Graph Neural Networks. KDD. doi: 10.1145/3711896.3736559.

[77] Lin, H., Yan, M., Ye, X., et al. (2023). A Comprehensive Survey on Distributed Training of Graph Neural Networks. Proceedings of the IEEE. doi: 10.1109/JPROC.2023.3337442.

[78] Joshi, C. K. (2025). Transformers are Graph Neural Networks. arXiv:2506.22084.

[79] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe, M. (2019). Weisfeiler and Leman Go Neural: Higher-Order GNNs. AAAI.

[80] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric Deep Learning: Going beyond Euclidean Data. IEEE Signal Processing Magazine. doi: 10.1109/MSP.2017.2693418.

[81] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. IEEE TNNLS. doi: 10.1109/TNNLS.2020.2978386.

[82] Zhang, Z., Cui, P., & Zhu, W. (2020). Deep Learning on Graphs: A Survey. IEEE TKDE. doi: 10.1109/TKDE.2020.2981333.

[83] Boll, H. O., Amirahmadi, A., Ghazani, M. M., et al. (2024). Graph neural networks for clinical risk prediction based on electronic health records: A survey. Journal of Biomedical Informatics. doi: 10.1016/j.jbi.2024.104616.

[84] Tang, J., Zhu, T., Zhou, W., et al. (2026). Graph neural networks for fMRI functional brain networks: A survey. Neural Networks. doi: 10.1016/j.neunet.2025.108137.

[85] Brussee, S., Buzzanca, G., Schrader, A. M. R., et al. (2025). Graph neural networks in histopathology. Medical Image Analysis. doi: 10.1016/j.media.2024.103444.

[86] Bo, D., Wang, X., Shi, C., & Shen, H. (2021). Beyond Low-frequency Information in Graph Convolutional Networks (FAGCN). AAAI.

[87] Pei, H., Wang, B., Lei, J., Yang, J., & Zhao, Y. (2021). Geom-GCN: Geometric Graph Convolutional Networks. ICLR.

[88] Liu, R., Xing, P., Deng, Z., et al. (2025). Federated Graph Neural Networks: Overview, Techniques, and Challenges. IEEE TNNLS. doi: 10.1109/TNNLS.2024.3360429.

[89] Xia, R., Liu, H., Li, A., et al. (2025). Incomplete graph learning: A comprehensive survey. Neural Networks. doi: 10.1016/j.neunet.2025.107682.

[90] Mujkanovic, F., Geisler, S., Günnemann, S., & Bojchevski, A. (2023). Are Defenses for Graph Neural Networks Robust? arXiv:2301.13694.

[91] Zügner, D., Akbarnejad, A., & Günnemann, S. (2018). Adversarial Attacks on Neural Networks for Graph Data (Nettack). KDD.

[92] Heid, E., Greenman, K. P., Chung, Y., et al. (2023). Chemprop: A Machine Learning Package for Chemical Property Prediction. Journal of Chemical Information and Modeling. doi: 10.1021/acs.jcim.3c01250.

[93] Choudhary, K., & DeCost, B. (2021). Atomistic Line Graph Neural Network for improved materials property predictions (ALIGNN). npj Computational Materials. doi: 10.1038/s41524-021-00650-1.

[94] Lin, K., Wang, L., & Liu, Z. (2021). Mesh Graphormer. ICCV.

[95] Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph Neural Networks in Recommender Systems: A Survey. ACM Computing Surveys.