Journal of Multiscale Material Informatics
ISSN 3047-5724

Research Article

Graph Neural Networks in Bioinformatics: From Biological Graph Construction to Interpretable and Scalable Precision Medicine

Seyed Enayatallah Alavi
Department of Computer Engineering, Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran
*Corresponding Author: se.alavi@scu.ac.ir

Abstract
Graph neural networks (GNNs) are increasingly used to integrate biological entities and relations across proteins, genes, drugs, molecular structures, and patients. This integrative narrative review critically examines how graph provenance, model architecture, evaluation design, explanation quality, and deployment constraints jointly determine the credibility of GNN findings in bioinformatics. In contrast to primarily architecture- and application-oriented surveys, we synthesize empirical case studies and propose a task-to-graph-to-validation reporting framework. We compare GCN, GAT, GraphSAGE, geometric/equivariant models, graph transformers, and graph–sequence hybrids; examine study-level benchmarks and baselines; and distinguish random, scaffold, protein-disjoint, temporal, and external validation. We also assess multi-omics fusion, graph-specific explanations, and computational scaling. Published examples show useful task-specific performance but do not justify universal superiority claims. We conclude that prospective utility depends on leakage-resistant evaluation, traceable graph construction, faithful and stable explanations, and independent biological or clinical verification.

Keywords: Graph neural networks; Bioinformatics; Multi-omics; Protein interactions; Drug discovery; Precision medicine.

Received: 11 October 2026 / **Revised:** 22 Ocotober 2026 / **Accepted:** 26 October 2026 / **Published:** 30 Ocotober 2026

© 2026 by the authors. This publication is licensed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. INTRODUCTION

Bioinformatics increasingly involves datasets in which the most informative signal is relational rather than independent. Protein–protein interaction (PPI) networks, gene regulatory networks, metabolic pathways, molecular interaction graphs, disease–gene networks, drug–target networks, and heterogeneous knowledge graphs all encode biological entities together with relationships among them. Conventional deep learning methods such as convolutional neural networks and sequence models are highly effective when the input can be represented on a regular grid or as an ordered sequence, but biological networks frequently violate those assumptions. Graph neural networks address this structural mismatch by learning representations through repeated aggregation and transformation of information associated with neighboring nodes.

The growing literature has established GNNs as a major approach to graph representation learning, while recent methodological reviews emphasize the breadth of their applications in life sciences. A recent survey of graph learning in bioinformatics organizes the field around graph construction, GNN architectures, and biological applications, reinforcing the importance of treating the graph as part of the scientific model rather than merely as an input format [1]. More general methodological primers similarly emphasize that GNNs learn functions over graph-structured data and that graph structure is particularly natural for life-science problems [2].

Deng et al. [1] survey architectures, graph construction, and application categories, whereas Corso et al. [2] provide a broad technical primer on graph machine learning in the life sciences. Those works make a conventional catalogue insufficient as a standalone contribution. The present review instead links each design choice to an auditable failure mode (edge uncertainty, entity overlap, batch dependence, explanation instability, or computational infeasibility) and to an appropriate control or validation test. This distinction is methodological rather than a claim that the underlying topics have not previously been reviewed.

The central question for bioinformatics is therefore not simply whether a GNN can obtain a high predictive score. A more consequential question is whether the graph encodes a defensible biological hypothesis and whether the learned representation generalizes beyond the graph and cohort used during training. This distinction matters because biological datasets are heterogeneous, incomplete, noisy, and frequently affected by selection bias. A model may exploit experimental artifacts, database density, study-specific conventions, or information leakage while appearing highly accurate under a random split.

This review addresses a different question from broad architecture catalogues [1], [2]: what evidence is needed to decide whether a GNN result is biologically credible and transportable? Its contributions are (i) an explicit graph-design decision framework linking nodes, edges, confidence, and heterogeneity to model choice (Table 1); (ii) cross-study empirical comparison with disclosed benchmarks, baselines, metrics, and limitations (Table 4); (iii) task-specific, leakage-aware validation guidance (Table 5); and (iv) connected assessment of graph explanations, multi-omics fusion, and computational scaling (Tables 5–6). These are methodological synthesis and reporting contributions, not claims of a new GNN architecture, systematic meta-analysis, or original wet-laboratory experimentation.

2. METHODS

Design and sources. This is a critical integrative narrative review, not a systematic review or meta-analysis. The initial manuscript reported PubMed, Google Scholar, and Web of Science as discovery sources, a 2018–2026 publication window, and 45 selected articles. However, no dated query exports, deduplication file, or article-level screening log survives in the supplied draft. Accordingly, the figure of 45 cannot be authenticated and is not treated as the number of studies analyzed. The supplied bibliography contained 18 entries. For this revision, their claims were checked where possible against publisher, PubMed, and conference records; targeted supplementary searches and source verification were carried out on 22 October 2026. The current reference list explicitly defines the citable evidence base, rather than reconstructing an undocumented flow diagram.

Transparent targeted update. Search concepts included ("graph neural network" OR GCN OR GAT OR GraphSAGE) AND (protein OR bioinformatics OR drug OR molecular OR multi-omics), supplemented by focused combinations of "scaffold split", "protein interaction", "graph explanation", "GNNExplainer", "PGExplainer", "equivariant", "GraphSAINT", and "graph transformer". Candidate records were located and checked using PubMed records, publisher articles, official conference proceedings, and benchmark documentation. Inclusion required identifiable publication metadata and substantive relevance to at least one of the review questions: graph formulation, comparative architecture, biological evaluation, explanation, fusion, or scalability. Empirical reports were prioritized for Table 4 where the original datasets, comparators, and metrics could be identified. General surveys were used for context, not counted as direct experimental demonstrations; non-GNN studies are labelled as contextual controls. Foundational methods published before 2018 were retained where indispensable [3], [5], [13], [34]. Excluded from study-level comparison were unverifiable performance assertions, secondary retellings without a recoverable original, and claims with missing benchmark definitions. No formal dual-reviewer screening, exhaustive database coverage, or risk-of-bias scoring is claimed.

Synthesis and extraction. Evidence was extracted narratively into fields for author/year, dataset, graph representation, architecture, target task, comparator, metric, key finding, and limitation. Differences across benchmark tasks, class imbalance, negative sampling, and split protocols precluded effect-size pooling. Source-reported quantities appear only with their task context; no experiments or independent re-runs were performed. A reproducible systematic search would require database-specific queries, export files, eligibility decisions for all hits, and a dated PRISMA-style flow, none of which can be reconstructed reliably from the original draft.

3. RESULTS AND DISCUSSION

A biological graph is defined as $G = (V, E, X, R, W, T)$, where V denotes entities, E observed relations, X node/edge attributes, R edge or entity types, W optional confidence weights, and T optional timestamps. Nodes can be genes, transcripts, proteins, drugs, compounds, patients, or samples. Edges can indicate physical binding, regulation, similarity, drug targeting, or measured co-occurrence. Every definition changes the prediction problem: a protein-node PPI graph predicts relations among known proteins; an atom/bond molecular graph predicts molecule-level outcomes; and a patient-similarity graph can allow cohort composition to influence sample predictions [1], [21]–[23].

Graph formation should therefore be prespecified as a modelling hypothesis. Node identity and granularity determine the prediction unit; edge semantics and direction determine which interactions can be propagated; confidence scores distinguish measured from weakly inferred relations; and attributes determine potential leakage channels. For example, Zitnik et al. represented proteins and drugs with several relation types rather than collapsing all edges into one adjacency matrix [21]. By contrast, Wang et al. constructed an omics-specific weighted patient-similarity graph for each measured view [23]. These designs solve different learning tasks and their results are not numerically comparable.

Three properties are particularly important. First, graph provenance should be recorded so that every edge can be traced to its evidence source when feasible. Second, uncertainty should not be hidden by converting weak or probabilistic associations into binary edges without qualification. Third, graph construction must respect the temporal and experimental boundaries of the prediction task. If an edge was

discovered using information that became available after the intended prediction time point, including it can create a form of temporal leakage.

Homogeneous graphs are simpler but can conflate biologically distinct relations. Heterogeneous graphs require type-specific transformations or relation-aware decoders for drug–target–disease contexts. Weighted graphs retain quantitative interaction confidence and permit reliability sensitivity analyses; binarization discards uncertainty. Dynamic graphs update edges across developmental stages, treatment courses, or database releases and require time-respecting feature engineering. Graph choice should be matched to the outcome, observation date, available evidence, and inductive or transductive deployment setting (Table 1).

Graph sparsity is another major issue. Some biological interaction maps are incomplete, and absence of an observed edge rarely means absence of a biological relationship. Consequently, link prediction tasks must distinguish between confirmed negatives and unknown relationships. Randomly sampling unobserved pairs as negatives can generate deceptively easy or biologically invalid benchmarks. More rigorous evaluation should consider experimentally supported negatives, temporal splits, degree-aware sampling, or external validation.

Table 1. Biological graph design decisions and appropriate model/evaluation responses

Graph design	Representation and typical task	Main risk	Model and audit implication
Homogeneous / static	One node and edge type; PPI functional label prediction	Interaction incompleteness; hub-driven bias	GCN or GraphSAGE for local propagation; degree-stratified negatives and protein-level holdout [3], [5], [34].
Heterogeneous / typed	Drug–protein–side-effect graph; relation-specific link prediction	Conflated edge semantics or target leakage	Relational GNN/typed decoders; report per-relation AUROC/AUPRC and held-out entities [21].
Weighted / uncertain	Confidence-weighted evidence edges or patient similarity	Weak evidence treated as certainty; batch-mediated similarity	Sensitivity to edge thresholds, source ablation and confidence calibration [23].
Dynamic / temporal	Time-stamped discoveries, treatments, or disease progression	Future information in edges/features	Time-respecting graph snapshots and external post-cutoff evaluation [19], [20].
Geometric / 3D	Residue and atom coordinates; molecular properties or interfaces	Coordinate noise, chirality and conformational shift	Equivariant or vector-valued models; structure/complex-level holdout [27], [28].

3.1. GNN Architectures Relevant to Bioinformatics

3.1.1. Graph Convolutional Networks

Graph convolutional networks (GCNs) extend the intuition of convolution to irregular graph structures. Kipf and Welling introduced a widely used GCN formulation in which node representations are updated using normalized adjacency information and trainable transformations [3]. In simplified form, a layer can be written as $H^{(l+1)} = \sigma(\check{D}^{-1/2} \hat{A} \check{D}^{-1/2} H^{(l)} W^{(l)})$, where $\hat{A}$ includes self-connections, $\check{D}$ is the corresponding degree matrix, $W^{(l)}$ is a trainable parameter matrix, and σ is a nonlinear activation. The important biological implication is that a node representation becomes informed by its local neighborhood.

GCNs use a shared, degree-normalized weighted sum: for a sparse graph and hidden width d, a typical layer costs approximately $O(|E|d + |V|d^2)$, excluding data loading. Their fixed, mainly isotropic aggregation is a useful low-parameter baseline for homophilous or moderately smooth biological signals, but cannot inherently distinguish relation types or assign context-specific significance to distinct neighbors. Deep stacks increase receptive fields but can over-smooth node representations. Uncertain edges, signed regulation, and highly heterophilous biological networks require feature engineering or more expressive message functions [2], [2].

3.1.2. Graph Attention Networks

Graph attention networks (GATs) introduce attention-based neighborhood aggregation so that a node can assign different weights to different neighbors [4]. This is appealing in biology because not all interactions are equally informative. Attention can provide a data-driven mechanism for weighting neighboring entities, although attention weights should not automatically be interpreted as causal or mechanistic explanations. The original GAT work demonstrated inductive and transductive capabilities and included protein–protein interaction data among its benchmarks [4].

Unlike fixed aggregation, GAT learns neighbor weights conditioned on the connected node features, potentially useful when interactions vary in local predictive relevance [4]. Sparse multi-head attention adds an edge-wise scoring term approximately O(|E|d) per head (in addition to projections and aggregation), with extra attention coefficients and memory. It does not automatically correct unreliable edges and does not guarantee better prediction than GCN on small, noisy cohorts. An attention coefficient is a model-internal weight, not evidence of causal influence; counterfactual perturbations are required before interpreting it biologically [4], [25], [26].

3.1.3. Inductive Neighborhood Aggregation and GraphSAGE

GraphSAGE learns a function that generates node representations by sampling and aggregating information from local neighborhoods rather than learning a separate embedding for every node [5]. Its inductive design is valuable for biological settings in which new nodes, samples, proteins, or graphs may appear after training. The original work demonstrated generalization to previously unseen nodes and graphs, including protein–protein interaction graphs [5].

GraphSAGE is expressly inductive: it learns an aggregation function applied to sampled neighborhoods so unseen nodes can be embedded using available attributes and neighbors [5]. A fixed fanout f at L layers may require O(B f^L) sampled node occurrences for B seed nodes in an unshared computation tree, illustrating the depth–variance trade-off. It is attractive for expanding PPI networks or graphs whose nodes are added after training, provided new-node features are available. GCN is often an efficient baseline on a fixed graph; GAT adds adaptive weighting when there is sufficient data to learn it; GraphSAGE is preferable when inductive scale and incremental nodes dominate. None is intrinsically a substitute for explicit relation or 3D geometry modelling [2]–[5].

3.1.4. Beyond Basic Message Passing

Beyond isotropic message passing, geometric and equivariant networks represent 3D coordinates and orientation-sensitive features. Geometric Vector Perceptron networks combine scalar and vector channels to learn from protein structures [28]; E(n)-equivariant networks preserve prescribed symmetries under coordinate transformations and have been demonstrated for molecular property prediction [27]. These approaches can exploit structural priors that 2D PPI graphs omit, but depend on conformer quality, structural coverage, and carefully defined symmetry assumptions. They are not automatically superior for sparse disease networks without spatial coordinates.

Graph transformers such as Graphormer encode structural position into attention to capture more distant dependencies than shallow local message passing [29]. However, dense all-pairs attention has $O(|V|^2)$ attention storage and quadratic pairwise work, often impractical for whole-genome graphs without sparse or hierarchical approximation. Hybrid graph–sequence models combine molecular connectivity or protein-residue graphs with sequence encoders, as illustrated by GraphDTA (drug molecular GNN plus protein sequence CNN) [22]. AlphaFold 2 and AlphaFold 3 [6,30] establish the importance of protein or complex structure prediction but are not controlled evidence that a GNN outperforms another model on downstream bioinformatics tasks. Comparative evaluation should hold split, features, parameters, and task constant wherever possible.

Table 2. Major GNN families and their relevance to bioinformatics

Model family	Core aggregation / approximate cost	Task-specific fit	Limitations / audit
GCN [3]	Degree-normalized fixed neighbor sum; $O(\|E\|d+\|V\|d^2)$ per layer	Fixed-graph, attribute-rich homogeneous biology	Simple baseline; limited relation sensitivity; over-smoothing
GAT [4]	Learned per-edge attention; added multi-head scoring and memory	Local interaction weighting, variable neighbor relevance	Attention not explanation; noise and parameter sensitivity

GraphSAGE [5]	Sampled mean/pooling/LSTM neighbor aggregation; inductive	Evolving PPI network, previously unseen nodes	Sampling variance and exponential fanout with depth
Typed GNN [21]	Edge-type-specific messages and decoders	Drug–protein–polypharmacy multi-relational prediction	Label sparsity and difficult relation-specific negatives
Graph transformer [29]	Structural encoding plus global attention, often $O(\|V\|^2)$	Molecular graph with non-local dependencies	Dense attention memory; controls needed for capacity
Geometric / EGNN [27], [28]	Coordinate-aware scalar/vector or equivariant updates	Protein and molecular structure-based tasks	Structure availability, conformer quality and geometric cost
Graph–sequence hybrid [22]	Molecular graph encoder coupled to protein sequence model	Drug–target affinity and combined evidence	Confounding by model size and mixed-modality overlap

3.2. Applications in Bioinformatics

3.2.1. Protein–Protein Interaction and Functional Prediction

Protein interaction networks are a natural application of GNNs because proteins can be represented as nodes and experimentally or computationally inferred interactions as edges. The original GAT study included a PPI benchmark, and subsequent work has developed graph-based approaches for interaction prediction and interface identification [4], [8]. A major advantage is that network context can complement protein-specific representations. Instead of treating each protein independently, the model can learn from its interaction neighborhood.

PPI benchmarking must distinguish held-out edges from held-out proteins. Zeng et al. [18] tested global GIN plus GIN-AK local subgraphs on SHS27K and SHS148K under random, breadth-first-search (BFS), and depth-first-search (DFS) partitions, reporting F1 improvements over several baselines. They also found multimodal feature changes affected F1 differently across those partitioning protocols (e.g., approximately 0.7%, 7%, and 4% gains on SHS27K under random, BFS, and DFS, respectively) [18]. These are within-split feature comparisons, NOT controlled differences in test-set difficulty, and BFS/DFS partitions should not be asserted to guarantee protein-disjointness without explicit verification. For a genuinely novel-protein claim, both endpoints or at least the held-out protein set must be disjoint from training as appropriate; protein-family holdouts provide an even stronger homology stress test [34].

3.2.2. Disease–Gene Association and Rare Disease Research

Disease–gene association can be formulated as link prediction or node classification in heterogeneous networks containing genes, diseases, phenotypes, pathways, and functional relationships. GNNs can integrate neighborhood information to identify candidate associations that would be difficult to infer from isolated features. This is especially relevant to rare diseases, where the available labelled cases may be limited and where network context can supply complementary evidence.

Nevertheless, rare disease applications are particularly sensitive to dataset imbalance and ascertainment bias. A scoping review of deep learning for rare disease research found that most studies focused on diagnosis and that image-based CNN approaches were common [9]. This broader literature highlights a general lesson for GNN studies: sophisticated architecture does not remove the need to characterize the underlying sampling process. For rare disease genomics, external validation and clinically meaningful error analysis are often more informative than marginal improvements in aggregate accuracy.

3.2.3. Drug Discovery and Drug–Target Interaction

Drug discovery contains several graph structures. A drug molecule can be represented as a molecular graph; a protein can be represented using residues, structural contacts, or interaction networks; and drug–target prediction can be formulated as a heterogeneous graph problem. GNNs are therefore positioned to combine chemical topology with biological context. Their value is strongest when the graph representation preserves meaningful chemistry or molecular structure rather than reducing a molecule to an arbitrary connectivity matrix.

Drug-target studies should not confuse an unseen pair with an unseen drug scaffold or protein. GraphDTA compared four graph encoders plus a protein-sequence CNN on Davis and KIBA, using concordance index and mean squared error against sequence-based deep-learning and classical baselines [22]. Its published benchmark results demonstrate competitive within-benchmark affinity prediction but do not, by themselves, establish simultaneous cold-drug and cold-target performance. Likewise, MoleculeNet and OGB standardize scaffold-based molecular property splits, which make chemically novel cores less

interchangeable with random compound splitting [19], [20]. External or prospective assays remain necessary for actual medicinal-chemistry transfer.

3.2.4. Multi-Omics Integration

Multi-omics inputs differ in distribution, dynamic range, sparsity, measurement error, dimension, and missingness. RNA-seq counts, methylation proportions, proteomic intensities, and metabolite abundances require modality-specific transformations rather than a shared unqualified z-score. Within the training fold, suitable operations may include library-size normalization and log transformation for counts, intensity normalization for proteomics, feature filtering, and quantile or variance stabilization where justified. Feature IDs should be mapped to a consistent gene/protein registry, duplicate identifiers resolved, and batch or site recorded as a covariate. Batch correction must be learned on training data and applied cautiously to test data; uncontrolled correction can remove true disease signal or incorporate information from the test cohort [10], [33].

Early fusion concatenates aligned omics into one feature vector or graph input; it is straightforward but vulnerable to high-dimensional dominance and complete-case attrition. Intermediate fusion encodes each modality separately, then exchanges information through cross-layer edges, co-attention, or learned graph embeddings; it can represent interactions but is harder to regularize. Late fusion combines calibrated prediction scores or label-space features after independent modality-specific learners, improving modularity and allowing modality-specific error analysis. MOGONET provides a concrete hybrid: each modality forms a patient-similarity graph and passes through a GCN, followed by a view-correlation discovery network that fuses label-space information [23]. Its published analyses used samples with matched omics, so performance under systematically missing modalities cannot be assumed. Missing views require explicit masks, a trained missing-modality strategy, and ablations at realistic missingness rates; absent measurements must not be imputed using held-out samples.

Table 3. Early, intermediate, and late fusion of heterogeneous omics

Fusion stag	Mechanism	Strength	Key weakness and t
Early	Normalize/align then concatenate into a sample vector or a shared graph	Simple training and unified representation	Feature-scale imbalance; require per-modality ablations and train-only feature selection.
Intermediate	Separate omics encoders connected through cross-layer edges, co-attention or shared latent nodes	Models interactions between molecular layers	Graph edge uncertainty, parameter cost, batch confounding; test cross-layer edge ablation.
Late (incl. MOGONET)	Independent view encoders; combine predicted probabilities or label-space dependencies [23]	Modular and interpretable per modality	Disagreement and calibration; matched-modality selection bias; assess masked-view and external cohorts.

3.2.5. Genomic Sequence Analysis and Graph–Transformer Integration

Sequence models and GNNs provide complementary inductive biases. Transformers can model long-range dependencies in DNA or RNA sequences, while GNNs can encode explicit relations among genes, loci, regulatory elements, proteins, or other biological entities. Recent reviews of transformer-based models for nucleotide sequence analysis document rapid growth in this area [11]. A promising direction is therefore to couple sequence representations with graph context rather than treating sequence and network information as competing alternatives.

Graph–sequence integration should be judged by controlled ablations: sequence-only, graph-only, fused models, comparable compute/parameter budgets, and entity-disjoint tests. DeepVariant [16] is an example of strong non-graph sequence-derived genomics modelling and illustrates why GNNs need not be the preferred tool for every genomics task. Applications also differ across organisms and assays, as reviewed in plant genomics [17]. AlphaFold 3 [30] uses diffusion-based modelling of biomolecular complexes and should be described as relevant structural context rather than misclassified as a conventional GNN. The precise comparison is between task-matched models, not broad technology families.

3.2.6. Precision Medicine and Clinical Decision Support

Precision medicine requires models that connect molecular evidence with phenotypes, diagnoses, treatments, and patient-specific characteristics. Heterogeneous graphs can represent these relationships and potentially support risk prediction, patient stratification, and treatment prioritization. Deep learning using genetic summary data illustrates the broader need to work under privacy and data-sharing constraints when individual-level genomic data cannot be freely distributed [12].

Clinical deployment imposes requirements beyond predictive discrimination. A clinically useful GNN should be calibrated, robust to distribution shift, transparent about uncertainty, and validated on populations that reflect the intended use. In addition, graph construction may itself contain protected or sensitive relationships. A model can therefore create privacy risks even if direct identifiers are removed, because relational structure may facilitate re-identification or inference.

Across the studies in Table 4, the task unit ranges from a patient to a protein, drug pair, or whole molecule; the graph and leakage unit therefore change together. Decagon's reported AUROC of 0.872 [21] cannot be meaningfully compared to the F1 reported for GNNGL-PPI [18] or concordance index for GraphDTA [22]. The repeated research gaps are more informative than cross-paper score ranking: weakly defined negatives, insufficiently independent held-out entities, incomplete modality coverage, and limited external experimental testing. These cross-study patterns motivate a task-specific reporting standard rather than a pooled claim of GNN improvement.

Table 4. Empirical comparison of representative biological graph studies (results are not pooled across tasks)

Study / dataset	Graph; architecture	Task; baselines	Metric; key finding	Critical limitation
Veličković et al. (2018) [4]; PPI benchmark	Protein interaction graph; multi-head GAT	Protein node multi-label classification; compared against GCN and other graph baselines	Micro-F1; attention-based aggregation competitive on PPI	Benchmark-specific node labels; not clinical or experimental validation.
Zitnik et al. (2018) [21]; multimodal drug/PPI network	Typed drug–drug, drug–protein, PPI relations; Decagon relational GCN	Polypharmacy side-effect relation prediction; RESCAL, DEDICOM, DeepWalk, concatenated drug features	Mean AUROC 0.872, AUPRC 0.832, AP@50 0.803 over 964 side-effect types	Retrospective annotated link prediction; no prospective adverse-event deployment.
Nguyen et al. (2021) [22]; Davis, KIBA	Atom/bond molecular graphs plus protein sequence; GCN/GAT/GIN /hybrid + CNN	Continuous drug–target affinity; DeepDTA, KronRLS and other baselines	Concordance index and MSE; graph representation improved task-specific benchmarks	Warm-pair benchmark cannot establish cold-scaffold or protein-family transfer.
Wang et al. (2021) [23]; ROSMAP, LGG, KIPAN, BRCA	Weighted sample-similarity graph per omics; GCN + VCDN (MOGONET)	AD classification, grade/subtype; NN_NN, NN_VCDN, MOGONET_NN	Accuracy, F1, AUC as applicable; fusion improved reported tasks	Analysis retained matched modalities; cross-site and missing-view robustness unproven.
Zeng et al. (2024) [18]; SHS27K, SHS148K	PPI global graph + local subgraphs; GIN and GIN-AK	Multi-category PPI types; RF, LR, PIPR, GNN-PPI, LDMGNN, SemiGNN-PPI	F1 on random/BFS/DFS partitions; reported improvements across protocols	Partition schemes are not proof of protein-family-disjoint validation.
Yang et al. (2019) [24]; 19 public	Molecular graph; directed	Molecular property	Task-dependent ROC-AUC/RMSE;	Endpoint heterogeneity

Study / dataset	Graph; architecture	Task; baselines	Metric; key finding	Critical limitation
and 16 proprietary molecular sets	message-passing network	prediction; descriptor/fingerprint-based and earlier graph models	strong benchmark performance on many datasets	and chemical split dependence limit universal ranking.

3.3. Critical Evaluation: Why Accuracy Alone Is Insufficient

The earlier broad statements that GNNs produce 10–15% or 15–20% improvements are removed because they lack a common benchmark, comparator, confidence interval, and split protocol. Valid empirical synthesis requires reporting outcome prevalence, evaluation unit, biological holdout, uncertainty estimates, baselines, and whether feature engineering used any held-out information. In particular, a random edge-level link split and a new-protein deployment question test different generalization targets. The validation design should follow the scientific claim, not merely maximize a leaderboard result [18], [20], [22], [34].

At minimum, GNN studies in bioinformatics should report discrimination metrics appropriate to the task, confidence intervals where feasible, class prevalence, calibration, and baseline performance. For imbalanced link prediction, area under the precision–recall curve may be more informative than accuracy or ROC-AUC alone. For clinical risk prediction, calibration and decision-curve analysis can provide information that discrimination metrics cannot. For molecular prediction, scaffold-aware or structure-aware splits are often more revealing than random splits.

Random splits estimate interpolation under roughly unchanged data and can inflate performance when homologous proteins, close chemical analogues, or related patients appear on both sides. Scaffold splits assign molecular core scaffolds to disjoint folds and interrogate chemical novelty, as operationalized in MoleculeNet and OGB [19], [20]. Protein-disjoint or homology-aware splits separate proteins or families rather than only edges, addressing identity overlap in PPI or drug–target tasks [34]. Temporal splits hold out discoveries after a fixed cutoff and require time-stamped graph snapshots, label availability, and metadata frozen before that cutoff. External validation assesses transport to independent laboratory, site, cohort, species, or assay conditions. These tests are complementary, not interchangeable. OGB, for example, uses scaffold partitions for molecular HIV/PCBA tasks and species-based partitions for PPI graph classification [20]. A temporal or external test is genuinely independent only if data acquisition and tuning are not adapted using those held-out labels.

Table 5. Leakage-resistant evaluation: what a split can and cannot establish

Strategy	Applicable example	Estimates	Major leakage to prevent	Implementation / reporting check
Random edge / sample	PPI edge labels or molecular affinity pairs	Interpolation to similar observations	Same protein, scaffold, patient, or replicated assay across folds	Deduplicate, group repeated observations; declare transductive message passing.
Scaffold-disjoint	Molecule property / drug discovery [19,20]	Chemical-core extrapolation	Scaffold overlap via salts, tautomer duplicates, near-identical series	Split chemical structures before scaling/selection; report scaffold overlap and scaffold definition.
Protein-disjoint / family	PPI and cold-target interaction [18,22,34]	Transfer to unseen proteins/families	Protein identity, high sequence homology, interaction partner leakage	Check endpoint and family overlap; keep derived features and graph edges fold-local where required.
Temporal	Future PPI discoveries or drug approvals	Forward-time generalization	Future graph edges, annotations,	Use dated snapshots and pre-cutoff

Strategy	Applicable example	Estimates	Major leakage to prevent	Implementation / reporting check
			database versions and tuning	normalization; report horizon and censoring.
External independent	New institution, assay, cohort or species	Transportability and distribution shift	Cross-site preprocessing, sample linkage and post-hoc selection	Freeze model and thresholds; disclose cohort shifts, confidence intervals and calibration.

3.4. Interpretability and Biological Meaning

Graph-specific attribution targets edges and connected motifs as well as node features. GNNExplainer optimizes a compact input subgraph and relevant features for an instance prediction [25], whereas PGExplainer learns a parameterized explanation generator applicable to multiple instances and inductive settings [26]. Edge masking, subgraph deletion, node feature perturbation, and counterfactual graph edits provide complementary post hoc tests. These differ from SHAP feature attribution [13]: biological meaning may reside in interacting node sets whose joint influence cannot be recovered by independent per-feature rankings. Attention coefficients should be reported as weights within the chosen model, not treated as automatically faithful explanations [4].

Evaluation must explicitly separate four properties. Explanation fidelity (faithfulness) asks whether retaining the highlighted subgraph preserves the model output and removing it substantially changes the prediction; suitable metrics include fidelity-positive/negative, sufficiency, comprehensiveness, and deletion curves. Stability measures agreement under repeated seeds, small graph perturbations, or resampled training subsets (e.g., Jaccard overlap of top-k edges). Biological plausibility asks whether highlighted motifs correspond to independent pathways, structural contacts, or external annotation rather than graph popularity. Experimental validation asks whether independent binding, knock-out/knock-down, or prospective perturbation experiments support a proposed mechanistic role. A model may score high on fidelity and low on biological truth: explanation of a prediction is not proof of a causal biological mechanism [14], [25], [26].

Explanations should be produced without test-label access, assessed against random and degree-matched motif baselines, and accompanied by uncertainty and replication. Perturbation should preserve chemically or biologically valid graph structure: deleting a covalent bond arbitrarily, for example, may create out-of-distribution molecules and make fidelity difficult to interpret. When possible, explanation precision should be compared with literature-curated interaction sets and then prospectively tested. Transparent architecture, reliable attribution, and validated mechanism are three different achievements.

3.5. Scalability, Reproducibility, and Deployment

Let $n=|V|$, $m=|E|$, d denote hidden width, and L the number of message-passing layers. For sparse GCN-style propagation, one layer requires approximately $O(md+nd^2)$ arithmetic and $O(m+nd)$ active graph/embedding storage, though training memory additionally stores activations for L layers (approximately $O(Lnd)$, implementation dependent). For dense graph attention, the attention matrix alone takes $O(n^2)$ storage, whereas sparse edge attention scales with m and number of heads. Full-batch training becomes expensive when biological graphs are huge or when repeated graph construction and feature matrices dominate RAM. Deeper layers propagate signals farther but increase activation memory and risk over-smoothing.

Under independent per-layer neighbor sampling, a seed batch of B nodes with fanout f across L layers can require $O(B f^{\wedge}L)$ sampled occurrences before deduplication; a numerical example with $B=512$, $f=10$ and $L=3$ is approximately 512,000 occurrences (not unique nodes or measured runtime). Layer-wise sampling bounds frontier growth only at the cost of sampling bias/variance and possible loss of rare biological neighbors. Cluster-GCN samples dense graph partitions to limit activation and neighborhood overhead, but may underrepresent cross-partition biological edges [31]. GraphSAINT samples and normalizes subgraphs, trading stochasticity against larger effective connectivity within mini-batches [32]. Distributed graph training can spread storage and compute but introduces communication, graph partitioning, security, and reproducibility overhead. Wall-clock comparisons require hardware, graph size, density, input feature width, batch size, precision, and sampling settings rather than architecture names alone.

Table 6. Complexity and scalability trade-offs for biological GNN workloads

Method	Dominant approximate cost / storage	Advantage for biological networks	Risk and necessary control
Sparse full-batch GCN	Per layer $O(md+nd^2)$, graph $O(m)$, activations about $O(Lnd)$	Uses complete observed PPI topology; stable neighborhood coverage	n/m growth, full embedding memory; report peak RAM and edges/sec.
Multi-head sparse GAT	Sparse edge aggregation + head-dependent edge attention ~$O(hm)$; $O(hm)$ coefficients	Adaptively weights neighbors	More memory; check performance vs simpler GCN under same parameter budget.
Neighbor sampling / GraphSAGE	Unshared worst-case sample expansion O(B f^L); bounded batch tensors	Inductive new nodes and very large interaction networks	Sampling variance, rare-neighbor loss; repeat seeds, document fanout [5].
Partition / Cluster-GCN	Subgraph-local compute, partition preprocessing	Better localized mini-batch training on large graphs	Lost boundary interactions, partition artifacts; test cut-edge sensitivity [31].
GraphSAINT subgraph sampling	Cost proportional to sampled edges/nodes, normalization overhead	Preserves connected mini-batch context	Sampling distribution shifts; report subgraph size and corrections [32].
Dense graph transformer	$O(n^2)$ attention work and storage plus projections	Global pairwise context for compact molecular graphs	Whole-network quadratic bottleneck; use sparse/hierarchical controls [29].
Distributed graph training	Partition-local compute + network synchronization	Massive knowledge graphs across compute nodes	Communication, privacy, synchronization variance; disclose network and hardware.

FAIR principles—findability, accessibility, interoperability, and reusability—are particularly relevant because biological graphs are often assembled from multiple public resources. A published model without a versioned graph specification may be impossible to reproduce even when the code is available. Thus, the graph itself should be treated as a first-class research artifact.

3.6. Ethical, Privacy, and Governance Considerations

Genomic and clinical graphs can encode sensitive relationships among individuals, families, diseases, and biological characteristics. Removing names or direct identifiers does not necessarily remove privacy risks because graph structure can contain unique patterns. Risk prediction using genetic data also raises questions about consent, secondary use, and equitable representation. These concerns are especially important when models are transferred between populations.

Responsible GNN research should therefore document data governance, consent constraints, access conditions, population composition, and intended use. When individual-level data cannot be shared, privacy-preserving learning, secure computation, federated approaches, or carefully designed summary-data methods may become relevant. The scientific objective should not be separated from governance: a model that cannot be responsibly validated or deployed may have limited translational value even if its benchmark performance is high.

3.7. Future Directions

First, graph–sequence and structure-informed models are promising but should be evaluated against strong sequence-only and geometric-only baselines. GraphDTA [22], GVP [28], Graphormer [29], and AlphaFold 3 [30] illustrate complementary representation strategies rather than a monotonic sequence of better methods. Future benchmarks should test molecular scaffold novelty, protein-family independence, structural uncertainty, and identical parameter/compute budgets.

Second, uncertainty-aware GNNs should distinguish uncertainty in graph edges, molecular assays, and predictive parameters. Confidence-weighted edges and ablation of low-confidence evidence can reveal whether a result relies on weak database associations. Calibration curves and out-of-distribution detection are particularly important when graph neighborhoods, assay batches, or patient demographics change.

Third, evaluation should move toward prospective and intervention-oriented benchmarks. A model that predicts an already well-characterized relationship is useful, but a stronger test is whether it identifies previously unknown associations that survive independent biological validation. Benchmark design should therefore reward novelty, calibration, robustness, and biological plausibility rather than only leaderboard performance.

Fourth, explainability should become an integrated part of model design. Rather than adding an explainer after training, future architectures can incorporate constraints or representations that make biological evidence more accessible. However, such mechanisms should be evaluated empirically for faithfulness and stability.

Finally, reproducible graph infrastructure is likely to become as important as model architecture. Versioned graph releases, provenance-aware edges, standardized benchmarks, entity-disjoint evaluation, and interoperable data schemas could make results more comparable across studies. Without these infrastructure improvements, rapid architectural innovation may continue to produce results that are difficult to reproduce or translate.

3.8. Limitations of the Present Review

This critical narrative review does not claim an exhaustive search, a preregistered protocol, a study-level quality score, or independent model reimplementation. The prior manuscript's statement that 45 articles were selected was not backed by an auditable record; only its 18-item reference list was visible, and the present expanded reference list should not be misconstrued as a retrospectively reconstructed screening outcome. Cross-study metrics are heterogeneous and should not be quantitatively pooled. Computational expressions are theoretical order-of-growth estimates, not measured timings on a standardized biomedical dataset. Stronger conclusions require externally validated comparative experiments, released split identifiers, and biological replication.

4. CONCLUSION

GNNs can integrate biological topology with molecular, patient, and clinical attributes, but the most useful model depends on the scientific task and on how the graph was built. The comparative evidence reviewed here supports a task-to-graph-to-validation framework rather than a universal accuracy hierarchy: GCN offers a simple local baseline, GAT adapts edge weights, GraphSAGE aids inductive scaling, typed GNNs respect relation semantics, and structural or graph–sequence hybrids exploit additional modalities under stronger data requirements. Published examples illustrate both capability and caution. Decagon predicted diverse polypharmacy relationships, MOGONET combined patient similarity across omics, and GNNGL-PPI used global and local interaction structure; yet their performance metrics and holdout units differ. Credible comparison requires explicit baselines, entity/scaffold/time-aware splitting, train-only graph preprocessing, calibration, graph explanation fidelity and stability, and where possible independent wet-laboratory or clinical testing. Taken together, the emerging direction is toward integrated graph–sequence–multimodal systems that can represent biological entities at multiple scales while remaining interpretable, scalable, and reproducible. If these methodological and governance challenges are addressed, GNNs can become an important component of precision bioinformatics rather than simply another high-capacity predictive model.

Data Availability

No new data were generated for this review. Publicly available datasets and databases discussed in the cited literature should be accessed through their original repositories and publications.

Conflict of Interest

The author declares no conflict of interest.

Author Contribution

Seyed Enayatallah Alavi conceptualized the review, conducted the literature synthesis, and wrote and revised the manuscript.

Funding
This work is supported in part by Shahid Chamran University of Ahvaz under grant number SCU.EC1401.450.

Acknowledgements
The author acknowledges Shahid Chamran University of Ahvaz for institutional support.

REFERENCES

[1] Deng, L., Dong, Z., Yang, Z., Gong, B., & Zhang, L. , 2026. Graph learning in bioinformatics: A survey of graph neural network architectures, biological graph construction and bioinformatics applications. Biomolecules, 16(2), 333. https://doi.org/10.3390/biom16020333
[2] Corso, G., Stark, H., Jegelka, S., Jaakkola, T., & Barzilay, R. , 2024. Graph neural networks. Nature Reviews Methods Primers, 4, 17. https://doi.org/10.1038/s43586-024-00294-7
[3] Kipf, T. N., & Welling, M. , 2017. Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations.
[4] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. , 2018. Graph attention networks. International Conference on Learning Representations. https://doi.org/10.17863/CAM.48429
[5] Hamilton, W. L., Ying, R., & Leskovec, J. , 2017. Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 30, 1024–1034.
[6] Jumper, J., Evans, R., Pritzel, A., et al. , 2021. Highly accurate protein structure prediction with AlphaFold. Nature, 596, 583–589. https://doi.org/10.1038/s41586-021-03819-2
[7] Sagendorf, J. M., Mitra, R., Huang, J., Chen, X. S., & Rohs, R. , 2024. Structure-based prediction of protein-nucleic acid binding using graph neural networks. Biophysical Reviews, 16, 297–314. https://doi.org/10.1007/s12551-024-01201-w
[8] Pancino, N., Gallegati, C., Bongini, P., et al. , 2024. Protein–protein interfaces: A graph neural network approach. International Journal of Molecular Sciences, 25(11), 5870. https://doi.org/10.3390/ijms25115870
[9] Lee, J., Liu, C., Kim, J., et al. , 2022. Deep learning for rare disease: A scoping review. Journal of Biomedical Informatics, 135, 104227. https://doi.org/10.1016/j.jbi.2022.104227
[10] Ballard, J. L., Wang, Z., Li, W., Shen, L., & Long, Q. , 2024. Deep learning-based approaches for multi-omics data integration and analysis. BioData Mining, 17, 38. https://doi.org/10.1186/s13040-024-00391-z
[11] Ghosh, N., Santoni, D., Saha, I., & Felici, G. , 2025. A review on the applications of Transformer-based language models for nucleotide sequence analysis. Computational and Structural Biotechnology Journal, 27, 1244–1254. https://doi.org/10.1016/j.csbj.2025.03.024
[12] Wang, A., Xiao, E., Cheng, J., & Shen, X. , 2026. Genetic risk predictions using deep learning models with summary data. Frontiers in Bioinformatics, 5, 1657021. https://doi.org/10.3389/fbinf.2025.1657021
[13] Lundberg, S. M., & Lee, S.-I. , 2017. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.
[14] Van Hilten, A., et al. , 2024. Designing interpretable deep learning applications for functional genomics: A quantitative analysis. Briefings in Bioinformatics, 25(6), bbae449. https://doi.org/10.1093/bib/bbae449
[15] El-Sayed, A. F., Darwish, S. M., Labib, N. M., et al. , 2026. Generative deep learning in bioinformatics: A microservices-driven paradigm for big data environment. Journal of Big Data, 13, 7. https://doi.org/10.1186/s40537-025-01320-5
[16] Poplin, R., Chang, P.-C., Alexander, D., et al. , 2018. A universal SNP and small-indel variant caller using deep neural networks. Nature Biotechnology, 36, 983–987. https://doi.org/10.1038/nbt.4235
[17] Fan, W., Guo, Z., Wang, X., Zhang, L., Liu, Y., Cai, C., Zhang, K., & Cheng, F. , 2025. Deep learning applications advance plant genomics research. Horticultural Plant Journal, 11(5), 1791–1806. https://doi.org/10.1016/j.hpj.2025.08.004
[18] Zeng, X., Meng, F.-F., Wen, M.-L., Li, S.-J., et al. , 2024. GNNGL-PPI: Multi-category prediction of protein-protein interactions using graph neural networks based on global graphs and local subgraphs. BMC Genomics, 25, 406. https://doi.org/10.1186/s12864-024-10299-x
[19] Wu, Z., Ramsundar, B., Feinberg, E. N., et al., 2018. MoleculeNet: A benchmark for molecular machine learning. Chemical Science, 9, 513–530. https://doi.org/10.1039/C7SC02664A

[20] 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. Advances in Neural Information Processing Systems, 33. https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html
[21] Zitnik, M., Agrawal, M., & Leskovec, J., 2018. Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics, 34(13), i457–i466. https://doi.org/10.1093/bioinformatics/bty294
[22] Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D., & Venkatesh, S., 2021. GraphDTA: Predicting drug–target binding affinity with graph neural networks. Bioinformatics, 37(8), 1140–1147. https://doi.org/10.1093/bioinformatics/btaa921
[23] Wang, T., Shao, W., Huang, Z., et al., 2021. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. Nature Communications, 12, 3445. https://doi.org/10.1038/s41467-021-23774-w
[24] Yang, K., Swanson, K., Jin, W., et al., 2019. Analyzing learned molecular representations for property prediction. Journal of Chemical Information and Modeling, 59(8), 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237
[25] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J., 2019. GNNExplainer: Generating explanations for graph neural networks. Advances in Neural Information Processing Systems, 32. https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html
[26] Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., & Zhang, X., 2020. Parameterized explainer for graph neural network. Advances in Neural Information Processing Systems, 33. https://papers.neurips.cc/paper/2020/hash/e37b08dd3015330dcbb5d6663667b8b8-Abstract.html
[27] Satorras, V. G., Hoogeboom, E., & Welling, M., 2021. E(n) equivariant graph neural networks. Proceedings of Machine Learning Research, 139, 9323–9332. https://proceedings.mlr.press/v139/satorras21a.html
[28] Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., & Dror, R., 2021. Learning from protein structure with geometric vector perceptrons. International Conference on Learning Representations. https://mlanthology.org/iclr/2021/jing2021iclr-learning/
[29] Ying, C., Cai, T., Luo, S., et al., 2021. Do transformers really perform badly for graph representation? Advances in Neural Information Processing Systems, 34, 28877–28888. https://proceedings.neurips.cc/paper/2021/hash/f1c1592588411002af340cbaedd6fc33-Abstract.html
[30] Abramson, J., Adler, J., Dunger, J., et al., 2024. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature, 630, 493–500. https://doi.org/10.1038/s41586-024-07487-w
[31] 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. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3292500.3330925
[32] Zeng, H., Zhou, H., Srivastava, A., Kannan, R., & Prasanna, V., 2020. GraphSAINT: Graph sampling based inductive learning method. International Conference on Learning Representations. https://mlanthology.org/iclr/2020/zeng2020iclr-graphsaint/
[33] Korsunsky, I., Millard, N., Fan, J., et al., 2019. Fast, sensitive and accurate integration of single-cell data with Harmony. Nature Methods, 16, 1289–1296. https://doi.org/10.1038/s41592-019-0619-0
[34] Park, Y., & Marcotte, E. M., 2011. Revisiting the negative example sampling problem for predicting protein–protein interactions. Bioinformatics, 27(21), 3024–3028. https://doi.org/10.1093/bioinformatics/btr514