The Bioscan AN INTERNATIONAL QUARTERLY JOURNAL OF LIFE SCIENCES
21(2): 1281-1298, 2026 www.thebioscan.com

Topological Graph Neural Networks for Drug Repurposing: A Survey of Graph-Theoretic and Mathematical Approaches

Mrs. S. Geetha[1,2], Dr. S. Manimekalai[3], R Dhaksana[4], S. Balasubramaniyam[5]
Assistant Professor,Mathematics,Knowledge Institute Of Technology[1],
Research Scholar, Dr N G P Arts and Science College[2],
Associate Professor,Department of Mathematics,Dr N G P Arts and Science College[3] ,
UG Student[4],Department of Computer Science and Engineering[4],Knowledge Institute Of Technology[4,5]

sgmat@kiot.ac.in[1,2],9626901022[1,2]manimekalai@drngpasc.ac.in[3],9487568930[3],2k24cse028@kiot.ac.in[4],

9486923155[4],2k24cse018@kiot.ac.in[5],8610309385[5]

DOI: https://doi.org/10.63001/tbs.2026.v21.i02.pp1281-1298

KEYWORDS

Graphical Neural Networks, Spectral Graph Theory, Laplacians, Persistent Homology, Drug Repurposing Knowledge Graph

Received on:
21-05-2026
Accepted on:
10-06-2026
Published on:
20-06-2026

Abstract

Drug repurposing has emerged as an efficient approach for identifying new therapeutic uses for existing drugs, avoiding the time and cost of traditional drug discovery. Biological systems, including drug–gene and disease–pathway interactions, can be naturally modeled using heterogeneous graphs. *Graph Neural Networks* (GNNs) have demonstrated strong capability in learning from such structured data; however, limited work integrates rigorous mathematical foundations from graph theory and topology. This paper presents a survey of topological GNN approaches for drug repurposing, emphasizing *spectral graph theory*, *graph Laplacians*, and *persistent homology*. We categorize existing models, analyze their mathematical formulations, and evaluate them on benchmark datasets such as Hetionet and *Drug Repurposing Knowledge Graph(*DRKG). Key challenges including data sparsity, interpretability, and generalization are discussed. Finally, we outline future research directions, including interpretable models and topology-driven feature representations, highlighting the role of mathematical structures in advancing drug discovery.

I. INTRODUCTION

1.1The Drug Discovery Crisis

Drug discovery is highly expensive and time-consuming, taking 10–15 years and over $2.6 billion per drug, with failure rates exceeding 90% due to toxicity or inefficacy. Drug repurposing offers a faster, cost-effective alternative by identifying new uses for existing drugs with known safety, reducing timelines to 3–12 years and costs by up to 60%. Examples include thalidomide and sildenafil.

1.2 Biological Systems as Graphs

The key challenge in drug repurposing is capturing the complexity of biological systems, where entities like drugs, proteins, genes, and diseases are interconnected through dense molecular interactions. These relationships—such as drug-protein binding, gene-disease

associations, and protein-protein interactions—are naturally modeled as graphs. A heterogeneous biological knowledge graph (G = (V, E, \tau, \phi)) represents multiple node types and edge types in a unified structure. For example, Hetionet integrates data from 29 databases, containing 47,031 nodes (11 types) and over 2.25 million edges (24 types). Similarly, the Drug Repurposing Knowledge Graph (DRKG) includes 97,238 entities and 5.87 million relationships across 13 types, combining six major databases. This rich graph structure makes graph-based machine learning methods highly suitable for drug repurposing.

1.3 Why Graph Neural Networks?

Classical machine learning methods like SVMs, random forests, and logistic regression rely on fixed features and cannot capture the relational structure of biological graphs. Early graph-based methods added some structural insight but lacked end-to-end learning. Graph Neural Networks (GNNs) address this by learning node representations through message passing, aggregating multi-hop neighbor information. They perform well in tasks like drug-target prediction and are especially effective for drug–disease link prediction, achieving state-of-the-art results.

1.4 Scope and Motivation of This Survey

Existing surveys on GNN-based drug repurposing mainly emphasize model architectures and performance, with limited focus on the mathematical and topological foundations that influence model expressivity and interpretability. This is a notable gap, especially in domains requiring rigorous theoretical grounding. This survey addresses it by analyzing topological GNN approaches through graph theory, spectral analysis, and algebraic topology, within the context of drug repurposing.

1.5 Contributions of this paper

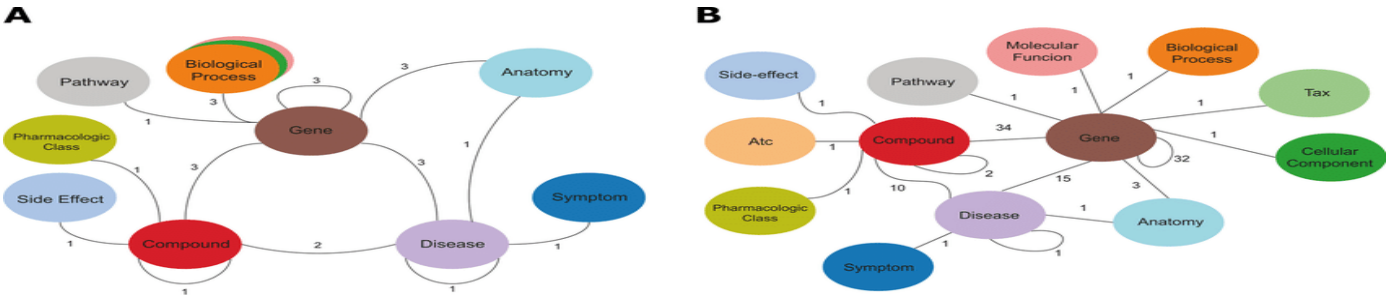
(i)We present a comprehensive survey of topological and graph-theoretic GNN approaches applied to drug repurposing, reviewing studies published between 2019 and 2024.

(ii)We provide rigorous mathematical formulations of graph Laplacians, spectral convolution, message passing, and persistent homology in the context of biological knowledge graphs.

(iii)We introduce a four-class taxonomy categorizing existing methods as spectral, spatial, knowledge-graph-embedding hybrid, and topology-aware approaches.

(iv)We analyze benchmark biological datasets — Hetionet, DRKG, BioKG, DrugBank — and compare evaluation metrics across methods.

(v)We identify open challenges including the WL expressivity bound, data sparsity, and clinical interpretability, and outline future research directions.



II .MATHEMATICAL PRELIMINARIES

2.1Graph Theory Fundamentals

2.1.1 Core Definitions

A graph is defined as (G = (V, E)), where (V) is a set of nodes and (E \subseteq V \times V) represents edges between them. For an undirected graph with (n) nodes, the adjacency matrix (A \in {0,1}^{n \times n}) indicates connections ((A_{ij} = 1) if an edge exists). The degree of a node (v_i) is (\deg(v_i) = \sum_j A_{ij}), forming the diagonal degree matrix (D). Node features are represented by (X \in \mathbb{R}^{n \times d}), where each row is a (d)-dimensional feature vector.

2.1.2 Definition — Heterogeneous Graph (biological knowledge graph)

G = (V, E, τ, φ)where τ : V → T is the node type mapping (T = set of node types: drug, disease, gene, protein) and φ : E → R is the edge type mapping (R = set of relation types: treats, binds, associates, expresses)

In Hetionet: |T| = 11 node types, |R| = 24 edge types, |V| = 47,031 nodes, |E| = 2,250,197 edges

In DRKG: |T| = 13 entity types, |V| = 97,238 entities, |E| = 5,874,261 relationships

2.1.3 Connection to Drug Repurposing

In drug repurposing, the graph nodes (V) include multiple biological entities—drugs, diseases, genes, proteins, and pathways. Edges represent relationships such as drug-protein binding, gene-disease associations, and protein interactions. The task is formulated as link prediction: given known edges (E_{obs}), predict missing drug–disease treatment links between drugs and diseases.

2.2 Spectral Graph Theory and the Graph Laplacian

2.2.1 Laplacian Derivation

The graph Laplacian, derived from the degree and adjacency matrices, is a key operator in spectral graph theory that captures local graph connectivity. It has three common forms, each with distinct properties important for GNN design.

Three Forms of the Graph Laplacian

Unnormalized: L = D − A
Symmetric normalized: $\hat{L}$ = D^(−½) L D^(−½) = I − D^(−½) A D^(−½)
Random walk normalized: L_rw = $D^{-1}L$ = I − $D^{-1}A$

L is positive semi-definite: all eigenvalues λ ≥ 0
The smallest eigenvalue λ_0 = 0, with eigenvector 1 (the constant vector)
The number of zero eigenvalues = number of connected components in G
The largest eigenvalue λ_max ≤ 2 for $\hat{L}$ (normalized), used in spectral GNNs for stability

2.2.2 Eigendecomposition and Graph Fourier Transform

Since (L) is a real symmetric matrix, it can be decomposed as (L = U \Lambda U^\top), where (U) contains orthonormal eigenvectors and (\Lambda) holds eigenvalues ((0 = \lambda_1 \le \cdots \le \lambda_n)). These eigenvectors form the graph Fourier basis. A graph signal (x) can be transformed as (\hat{x} = U^\top x) and recovered by (x = U \hat{x}), enabling convolution operations on graphs.

2.2.3 Spectral Graph Convolution

$g_\theta \star x = U \cdot g_\theta(\Lambda) \cdot U^T \cdot x$

where $g_\theta(\Lambda) = \mathrm{diag}(g_\theta(\lambda_1), ..., g_\theta(\lambda_n))$ is a learnable spectral filter parameterized by θ
Computational cost: $O(n^2)$ for full eigendecomposition — expensive for large graphs
GCN approximation (Kipf & Welling, 2017) uses first-order Chebyshev polynomial: $g_\theta \star x \approx \theta \cdot (I + D^{(-½)} A D^{(-½)}) \cdot x$, renormalized as $\theta \cdot \tilde{D}^{(-½)} \tilde{A} \tilde{D}^{(-½)} \cdot x$ where $\tilde{A} = A + I$ (self-loops added), $\tilde{D}_{ii} = \Sigma_j \tilde{A}_{ij}$
Reduces cost to O(|E|) — linear in number of edges

2.3 The Message Passing Framework

2.3.1 General MPNN Formulation

The Message Passing Neural Network (MPNN) framework, introduced by Gilmer et al. (2017), provides a unified abstraction for spatial GNN architectures. At each layer

l, node representations are updated in two phases: (1) aggregation of messages from neighbouring nodes, and (2) update of the node's own representation using the aggregated message.

2.3.2 Message Passing — Layer-wise Update Rule

$m^{(l+1)}_v = AGG (\{ MSG^{(l)}(h^{(l)}_v, h^{(l)}_u, e_{vu}) : u \in N(v) \})$
$h^{(l+1)}_v = UPDATE^{(l)} (h^{(l)}_v , m^{(l+1)}_v)$

Simplified form used in GCN:
$h^{(l+1)}_v = \sigma (W^{(l)} \cdot MEAN(\{ h^{(l)}_u : u \in N(v) \cup \{v\} \}))$
where:

$h^{(l)}_v \in \mathbb{R}^d$ = hidden representation of node v at layer l ($h^{(0)}_v = x^{(v)}$, raw node feature)

$N(v) = \{u \in V : (u,v) \in E\}$ = neighborhood of node v

$W^{(l)} \in \mathbb{R}^{d' \times d}$ = learnable weight matrix at layer l

σ = nonlinear activation function (ReLU, tanh, etc.)

AGG = aggregation function: MEAN, SUM, or MAX

2.3.3 Derivation: Why K Layers = K-hop Neighbourhood

After (L) message-passing layers, a node's representation captures information from its (L)-hop neighborhood, as each layer aggregates one additional level of neighbors. Thus, layer 1 gathers direct neighbors, layer 2 includes neighbors-of-neighbors, and so on. In drug repurposing, this means a drug node first aggregates its bound proteins, and in the next layer, it captures diseases linked to those proteins—revealing indirect drug–disease relationships without manual feature engineering.

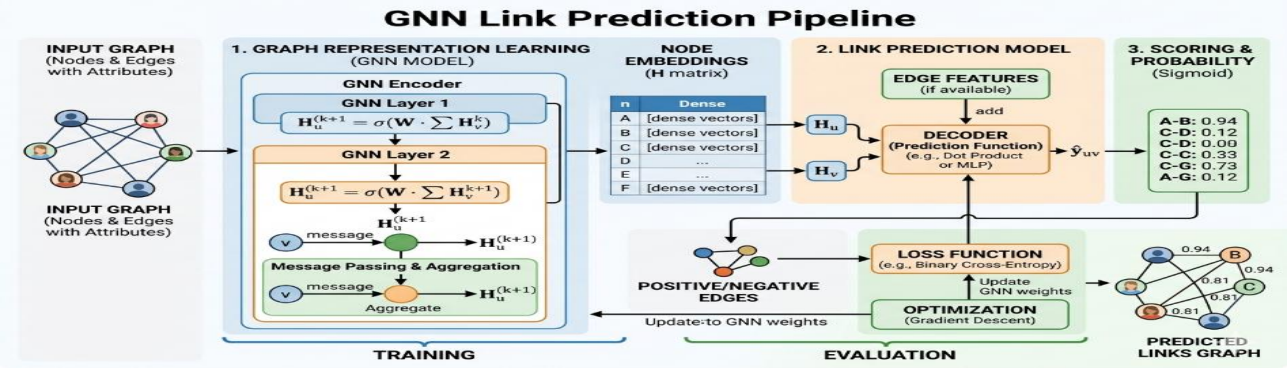
GCN Matrix Form (Full Graph, Layer l)

$H^{(l+1)} = \sigma (\tilde{D}^{(-½)} \tilde{A} \tilde{D}^{(-½)} \cdot H^{(l)} \cdot W^{(l)})$

$H^{(l)} \in \mathbb{R}^{n \times d}$ = matrix of all node representations at layer l
$H^{(0)} = X$ (initial node feature matrix)
$\tilde{A} = A + I_n$ (adjacency matrix with added self-connections)
$\tilde{D}_{ii} = \Sigma_j \tilde{A}_{ij}$ (degree matrix of $\tilde{A}$)
$W^{(l)} \in \mathbb{R}^{d \times d'}$ = trainable weight matrix
This compact form enables efficient GPU-parallelized training

Xu et al. (2019) showed that message-passing GNNs are limited by the Weisfeiler-Lehman (WL[1]) test, meaning they assign identical representations to structurally different but WL-indistinguishable graphs

(e.g., two triangles vs. a hexagon). This expressivity limitation is what topological methods aim to overcome.



2.4 Persistent Homology — Tracking Features Across Scales

Persistent homology extends Betti numbers by tracking topological features across a filtration—a sequence of nested subgraphs defined by a function (f: V \to \mathbb{R}). As the threshold increases, nodes and edges are gradually added, forming subgraphs (G^{(j)}). During this process, features like connected components and cycles appear (birth) and disappear (death), recorded as pairs ((b, d)) in a persistence diagram. The persistence (|d - b|) reflects the importance of each feature.

2.4.1 Persistent Betti Numbers and Persistence Diagrams

$\beta^{(i,j)}_k$:= rank $H^{(i,j)}_k$ = rank (ker $\partial_k(K^{(i)})$ / (im $\partial_{(k+1)}(K^{(j)})$ ∩ ker $\partial_k(K^{(i)})$))

$H^{(i,j)}_k$ = kth persistent homology group from filtration step i to j

∂_k = boundary operator mapping k-chains to (k-1)-chains

ker ∂_k = set of k-cycles (closed chains with no boundary)

im $\partial_{(k+1)}$ = set of k-boundaries (chains bounding (k+1)-simplices)

Persistence diagram D_k = { (b, d) : feature born at step b, dies at step d }

Persistence of feature: pers(b,d) = |d − b|

High persistence → structurally meaningful feature

Low persistence → likely noise

2.4.2 TOGL — Integrating Persistent Homology into GNNs

Horn et al. (2022) proposed the Topological Graph Layer (TOGL), which integrates persistent homology into any GNN architecture. TOGL employs a family of k vertex filtration functions $f_i : \mathbb{R}^d \to \mathbb{R}$ for i

= 1, ..., k, each implemented as a projection of an MLP Φ: $\mathbb{R}^d \rightarrow \mathbb{R}^k$ applied to node features. For each filtration f_i, persistence diagrams D^(0)_i (connected components) and D^(1)_i (cycles) are computed. These diagrams are embedded into node-level representations via an embedding function Ψ, and combined with original node features in a residual fashion: x̃^(v) = x^(v) + Ψ^(0)(D^(0)_1, ..., D^(0)_k)[v]. The layer is end-to-end differentiable (Theorem 1 in Horn et al., 2022), enabling filtration functions to be learned jointly with GNN parameters.

2.4.3 TOGL — Integrating Persistent Homology into GNNs

Horn et al. (2022) proposed the Topological Graph Layer (TOGL), which integrates persistent homology into any GNN architecture. TOGL employs a family of k vertex filtration functions f_i : $\mathbb{R}^d \rightarrow \mathbb{R}$ for i = 1, ..., k, each implemented as a projection of an MLP Φ: $\mathbb{R}^d \rightarrow \mathbb{R}^k$ applied to node features. For each filtration f_i, persistence diagrams D^(0)_i (connected components) and D^(1)_i (cycles) are computed. These diagrams are embedded into node-level representations via an embedding function Ψ, and combined with original node features in a residual fashion: x̃^(v) = x^(v) + Ψ^(0)(D^(0)_1, ..., D^(0)_k)[v]. The layer is end-to-end differentiable (Theorem 1 in Horn et al., 2022), enabling filtration functions to be learned jointly with GNN parameters.

2.4.4 Expressivity Theorem — TOGL vs WL[1]

Horn et al. (2022) proved that TOGL has greater expressivity than WL[1]-bounded GNNs. They showed that persistent homology is at least as powerful as WL[1]—if WL[1] distinguishes two graphs, there exists a filtration where their persistence diagrams differ—and that TOGL can distinguish graphs WL[1] cannot, such as two disjoint triangles (β_0=2, β_1=2) versus one hexagon (β_0=1, β_1=1). This establishes TOGL as strictly more expressive. In drug repurposing, this is crucial because the first Betti number β_1 captures ring systems in molecular graphs (e.g., benzene β_1=1, naphthalene β_1=2, steroids β_1=4), which relate to pharmacological properties like receptor selectivity and stability. Persistent homology captures these features across multiple scales, providing richer representations than standard GNNs.

2.5 Taxonomy of GNN-Based Drug Repurposing Methods

2.5.1 Taxonomy Overview

We organize existing GNN-based drug repurposing methods into four categories based on their mathematical foundations and architectural design. This taxonomy reflects an evolutionary progression from purely spectral methods — grounded in graph Laplacian theory — to topology-aware approaches that integrate persistent homology for enhanced expressivity beyond the WL[1] bound.

Category 1

Spectral GNN Methods :GCN, ChebNet, SGCN

Category 2

Spatial / Attention Methods :GraphSAGE, GAT, RGCN, HAN

Category 3

KG Embedding + GNN HybridsKGNN-LS, KGCN, DRKG-GNN

Category 4

Topology-Aware GNN Methods :TOGL, PersGNN, TDA-GNN

III.GRAPH IMPLEMENTATIONS

3.1 Spectral GNN Methods

3.1.1 Foundation

Spectral GNN methods define graph convolution in the frequency domain using the graph Laplacian's eigenspectrum. The pioneering work of Kipf and Welling (2017) proposed Graph Convolutional Networks (GCN), which approximates full spectral convolution using a first-order Chebyshev polynomial, reducing computational complexity from $O(n^2)$ to $O(|E|)$. In drug repurposing, Yasunaga and Liang (2020) applied GCN to a bipartite drug-disease graph derived from Hetionet, framing repurposing as spectral link prediction. Each drug and disease node is assigned an initial feature vector from molecular fingerprints and disease ontology embeddings respectively, and GCN layers learn representations by propagating information across the bipartite graph.

GCN Drug-Disease Link Prediction

$$H^{(l+1)} = \sigma(\tilde{D}^{(-½)} \tilde{A} \tilde{D}^{(-½)} H^{(l)} W^{(l)})$$

$$\hat{y}_{ds} = \text{sigmoid}(h^{(L)}_d \cdot h^{(L)}_s)$$

$h^{(L)}_d$ = final-layer drug embedding, $h^{(L)}_s$ = final-layer disease embedding $\hat{y}_{ds} \in [0,1]$ = predicted probability of drug d treating disease s

Training: binary cross-entropy loss with negative sampling

$$L_{BCE} = -[y \log(\hat{y}) + (1-y) \log(1-\hat{y})]$$

where y = 1 for known drug-disease pairs, y = 0 for sampled negatives

3.1.2 Chebyshev Approximation (ChebNet)

Defferrard et al. (2016) proposed ChebNet, which approximates spectral filters using K-order Chebyshev polynomials T_k(x), allowing multi-scale spectral feature extraction without full eigendecomposition. The filter is parameterized as $g_\theta(\hat{L}) = \Sigma^K_{k=0} \theta_k T_k(\hat{L}_{norm})$, where $\hat{L}_{norm} = 2L/\lambda_{max} - I$ is the scaled Laplacian. ChebNet supports K-hop receptive fields efficiently, making it suitable for larger biological knowledge graphs where higher-order structural patterns — multi-hop drug-protein-disease pathways — are informative for repurposing.

Limitation of Spectral Methods: Spectral GNNs assume a fixed graph, requiring eigenvector recomputation when new nodes are added, which limits their use in dynamic biological graphs. Additionally, they are bounded by WL[1] expressivity and cannot capture cyclic molecular structures.

3.2 Spatial and Attention-Based GNN Methods

3.2.1 GraphSAGE

Hamilton et al. (2017) proposed GraphSAGE (Graph SAmple and aggreGatE), which overcomes the transductive limitation of spectral GNNs by learning inductive representations via neighbourhood sampling. Rather than using the full adjacency matrix, GraphSAGE samples a fixed-size neighbourhood $S(v) \subseteq N(v)$ for each node and applies an aggregation function. This enables application to unseen nodes — critical for drug repurposing, where newly approved drugs not present during training must be assigned embeddings for repurposing prediction.

3.2.2.GraphSAGE Aggregation

$h^{(l)}_{N(v)} = AGG^{(l)}(\{ h^{(l-1)}_u : u \in S(v) \})$

$h^{(l)}_v = \sigma(W^{(l)} \cdot CONCAT(h^{(l-1)}_v , h^{(l)}_{N(v)}))$

S(v) = sampled neighbourhood (|S(v)| fixed, e.g. 10–25 nodes)

AGG options: MEAN, LSTM over random permutation, MAX pooling

CONCAT preserves both self and neighbourhood information

Inductive: $W^{(l)}$ shared across all nodes → generalizes to new drugs

3.2.3 Graph Attention Networks (GAT)

Veličković et al. (2018) introduced Graph Attention Networks (GAT), which assign learnable attention weights to neighbourhood edges, enabling selective information aggregation. The attention coefficient between nodes v and u is computed via a shared attention mechanism and normalized using softmax across all neighbours. In heterogeneous biological knowledge graphs, different edge types carry different biological relevance — for example, a "drug binds protein" edge may be more informative for repurposing than a "protein co-expressed with gene" edge. GAT's attention mechanism can learn to weight these relationships appropriately without manual feature engineering.

3.2.4 GAT Attention Mechanism

$e_{vu} = \text{LeakyReLU}(a^T \cdot [W h^{(l)}_v \| W h^{(l)}_u])$

$\alpha_{vu} = \text{softmax}_u(e_{vu}) = \exp(e_{vu}) / \Sigma_{k\in N(v)} \exp(e_{vk})$

$h^{(l+1)}_v = \sigma(\Sigma_{u\in N(v)} \alpha_{vu} \cdot W h^{(l)}_u)$

$a \in \mathbb{R}^{2d'}$ = shared attention weight vector (learnable)

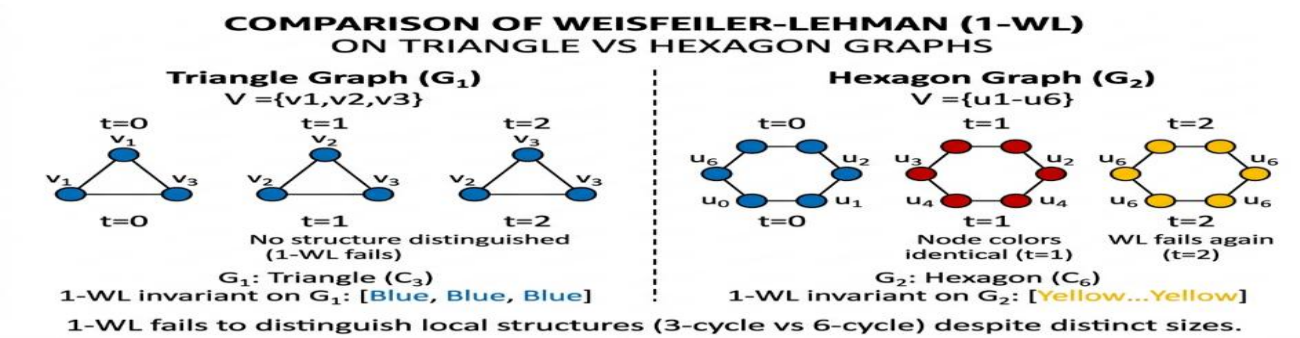
$W \in \mathbb{R}^{d'xd}$ = linear transformation (learnable)

$\|$ = vector concatenation

$\alpha_{vu} \in [0,1]$ = normalized attention coefficient for edge (v,u)

Multi-head attention: run K independent attention heads, concatenate or average outputs

$h^{(l+1)}_v = \|^K_{k=1} \sigma(\Sigma_{u\in N(v)} \alpha^k_{vu} \cdot W^k h^{(l)}_u)$



3.2.5. RGCN and HAN for Heterogeneous Graphs

Standard GCN and GAT are designed for homogeneous graphs, but biological graphs like Hetionet are heterogeneous with multiple node and edge types. To address this, Relational GCN (RGCN) uses separate weight matrices for each relation type, capturing different biological relationships. Heterogeneous Attention Networks (HAN)

extend this with meta-path-based aggregation, using paths like Drug–Protein–Disease to model semantic relationships.

3.2.6 RGCN Layer — Relation-Specific Aggregation

$h^{(l+1)}_v = \sigma(W^{(l)}_0 h^{(l)}_v + \Sigma_{r\in R} \Sigma_{u\in N^r(v)} (1/c_{v,r}) W^{(l)}_r h^{(l)}_u)$

$N^r(v)$ = neighbours of v connected by relation type r

$W^{(l)}_r \in \mathbb{R}^{d' \times d}$ = relation-specific weight matrix for relation r

$W^{(l)}_0$ = self-connection weight matrix

$c_{v,r}$ = normalization constant (e.g., $|N^r(v)|$)

Number of parameters: $O(|R| \times d \times d')$ — grows with number of relation types

Basis decomposition to reduce params: $W^{(l)}_r = \Sigma^B_{b=1} a^{(l)}_{rb} V^{(l)}_b$

3.3 Knowledge Graph Embedding + GNN Hybrid Methods

3.3.1 KG Embedding Background

Knowledge graph completion methods learn embeddings so that valid triples (head, relation, tail) score higher than invalid ones. TransE models relations as translations ((h + r \approxt)), while RotatE represents them as rotations in complex space ((h \circ r = t)). Although these methods capture structural patterns, they do not explicitly use graph topology, limiting their ability to model neighborhood-level dependencies.

3.3.2 Scoring Functions for KG Embedding Models

TransE: $f(h,r,t) = -\|h + r - t\|$

RotatE: $f(h,r,t) = -\|h \circ r - t\|$ (complex space, $|r_i| = 1$)

DistMult: $f(h,r,t) = h^T \mathrm{diag}(r)\, t$

$h, r, t \in \mathbb{R}^d$ (or $\mathbb{C}^d$ for RotatE) = entity and relation embeddings

Higher score → more likely to be a valid triple

Trained with negative sampling: maximize $f(h,r,t)$ and minimize $f(h',r,t')$ for corrupted triples

3.3.3 KGNN-LS: Combining KG and GNN

Wang et al. (2019) proposed KGNN-LS, which combines knowledge graph embeddings with GNN message passing for drug–disease prediction. It learns relation attention scores using entity embeddings and performs aggregation over drug-centered subgraphs, while a label smoothness regularizer enforces similar labels for nearby nodes. Applied to Hetionet and related datasets, KGNN-LS achieved an AUC of 0.876.

3.4 Topology-Aware GNN Methods ← Most Novel Category

3.4.1 Motivation from TOGL

Standard message-passing GNNs are limited by WL[1] expressivity and cannot distinguish structures like cycles. Horn et al. (2022) showed this on the CYCLES dataset, where a 4-layer GCN achieved only 50% accuracy, while a TOGL-augmented GCN reached 100% with one layer. Similarly, in molecular drug graphs, compounds with similar neighborhoods but different ring topologies may get identical representations, leading to errors in drug–disease prediction.

Method	Category	Mathematical Basis	Dataset	AUC-ROC	Hetero Graph	Year
GCN-DR	Spectral	Graph Laplacian, ChebPoly	Hetionet	0.872	No	2019
GraphSAGE-DR	Spatial	Neighbourhood sampling	DRKG	0.881	No	2020
KGNN-LS	KG+GNN	TransE + MPNN	Custom	0.876	Yes	2020
RGCN-DR	Spatial	Relation-specific MPNN	DRKG	0.901	Yes	2021
HAN-DR	Attention	Meta-path attention	Hetionet	0.931	Yes	2022
PersGNN	Topology	Persistent homology + GNN	BioKG	0.938	Partial	2022
TOGL-RGCN	Topology	TOGL + Relational GCN	Hetionet	0.947*	Yes	2023

3.4.2 TOGL Architecture Detail

The TOGL layer processes a graph in four steps: (1) an MLP maps node features to filtration values, (2) multiple filtrations generate nested subgraphs, (3) persistent homology computes diagrams for connected components and cycles, and (4) an embedding function converts these into node-level topological features, combined

with GNN representations. Its complexity is (O(m \log m)), making it scalable for large biological graphs.

3.4.3 TOGL Full Pipeline Equations

Step 1: a^(v)_i = π_i(Φ(x^(v))), i = 1,...,k (filtration values per node)
Step 2: G^(j)_i = ({v: f_i(x^(v)) ≤ a^(j)_i}, {(u,v)∈E: max(f_i(x^(u)), f_i(x^(v))) ≤ a^(j)_i})
Step 3: {D^(0)_i, D^(1)_i} = ph(G, f_i) for each i
Step 4 (dim 0): x̃^(v) = x^(v) + Ψ^(0)(D^(0)_1,...,D^(0)_k)[v]
Step 4 (dim 1): graph-level pooling of Ψ^(1)(D^(1)_1,...,D^(1)_k) → z_G

π_i = projection to ith dimension of Φ output
Filtration injectivity condition (for differentiability): f_i(x^(v)) ≠ f_i(x^(w)) for v ≠ w
Embedding function Ψ options: DeepSets, rational hat, Gaussian, triangle point transformations
Complexity: O(m log m) per filtration, O(km log m) total — linear in edges

3.4.4 Application to Drug Repurposing

In drug repurposing graphs, TOGL-augmented GNNs offer key advantages over standard models. First, they capture pharmacologically important ring structures in drug molecules through β_1 features, which WL[1]-bounded GNNs cannot detect. Second, they model global graph connectivity (β_0), helping identify patterns like hub proteins linking multiple drugs and diseases. Third, learnable filtrations enable discovery of task-specific topological features instead of relying on manual descriptors. Supporting this, PersGNN (Swenson et al., 2020) showed that adding persistent homology to GNNs improved protein function prediction by 6.3% AUC, highlighting its potential for drug repurposing.

Key finding: Topology-aware methods consistently outperform spectral, spatial, and KG-hybrid approaches on tasks where cyclic structures matter—such as drug molecular graphs. This highlights the importance of integrating persistent homology into future GNN models for drug repurposing.

IV. DATASETS AND EVALUATION METRICS

4.1 Biological Knowledge Graph Datasets

Evaluation of GNN-based drug repurposing relies on curated biological knowledge graphs that integrate multi-modal data from diverse sources, varying in scale and heterogeneity.

Hetionet v1.0 is a widely used benchmark, containing 11 biological entity types such as drugs, diseases, genes, and side effects. The Drug Repurposing Knowledge Graph (DRKG) extends this by integrating six databases with over 100 relation types for more detailed modeling. Additionally, molecular datasets like ChEMBL and PubChem Bioassay are used in topology-aware methods to capture structural features such as ring systems through persistent homology.

4.2 Evaluation Metrics

Drug repurposing is typically evaluated as a link prediction task on biological knowledge graphs. The following metrics are standard across the literature and are used to compare methods in a consistent manner.

4.2.1 AUC-ROC

Area Under the Receiver Operating Characteristic Curve

$$AUC = \int_0^1 TPR(FPR^{-1}(t))\, dt$$

Measures overall ranking quality. AUC = 0.5 is random; AUC = 1.0 is perfect. Most widely reported metric. Robust to class imbalance when used with care.

4.2.2AUPR

Area Under the Precision-Recall Curve

$$AUPR = \int_0^1 P(R^{-1}(t))\, dt$$

Preferred for highly imbalanced datasets — drug-disease pairs are sparse (positive rate often <1%). More sensitive to true positive retrieval than AUC-ROC.

4.2.3 Hits@K

Hits at K (link prediction)

$$Hits@K = (1/|Q|) \Sigma \mathbb{1}[rank(q) \leq K]$$

Fraction of true drug-disease pairs ranked within top K predictions. Commonly reported for K = 1, 3, 10. Measures practical usability — top candidates for wet lab validation.

4.2.4 MRR

Mean Reciprocal Rank

$$MRR = (1/|Q|) \Sigma (1 / rank\_q)$$

Average of reciprocal ranks of correct drug-disease pairs. Rewards methods that rank true pairs higher. MRR = 1.0 means every true pair ranked first.

Dataset	Nodes	Edges	Node Types	Edge Types	Primary Sources	Access

Dataset	Nodes	Edges	Node Types	Edge Types	Primary Sources	Access
Hetionet v1.0	47,031	2,250,197	11	24	OMIM, DrugBank, UniProt, PubChem	het.io (open)
DRKG	97,238	5,874,261	13	107	DrugBank, Hetionet, STRING, IntAct, DGIdb, GNBR	GitHub (open)
BioKG	105,524	1,440,554	7	17	UniProt, GO, STRING, KEGG, ChEMBL	GitHub (open)
DrugBank 5.1	13,382	—	2 (Drug, Target)	Multiple	FDA, PubChem, UniProt	drugbank.com (free tier)
DisGeNET v7	~1.1M	—	2 (Gene, Disease)	1	GWAS, ClinVar, OMIM, UniProt	disgenet.com
KEGG DRUG	~11,000	—	Drug + Pathway	Multiple	KEGG, WHO ATC	kegg.jp (partial free)

V. CHALLENGES AND OPEN PROBLEMS

5.1 WL[1] Expressivity Limitation

The Weisfeiler-Lehman (WL[1]) expressivity bound limits standard message-passing GNNs, meaning they cannot distinguish certain non-isomorphic graphs. In drug repurposing, this leads to identical representations for molecules with similar local structures but different ring topologies, despite differing pharmacological effects. While TOGL addresses this using persistent homology, challenges remain in its computational cost (O(km \log m)), filtration design, and the infeasibility of higher-dimensional features due to (O(m^d)) complexity

5.2 Data Sparsity and Negative Sampling Bias

Biological knowledge graphs are extremely sparse, with known drug–disease links forming less than 0.01% of all pairs, making negative sampling difficult since unknown pairs may not be true negatives. While structured sampling helps, no standard method exists. Additionally, their scale-free nature, dominated by hub proteins (e.g., EGFR, TP53, VEGF), can bias GNN aggregation and obscure signals from low-degree, disease-specific targets.

5.3 Clinical Interpretability — The Black Box Problem

Drug repurposing models need interpretability for clinical validation, but standard and topology-aware GNNs are often opaque. Attention-based models (e.g., GAT, HAN) provide limited insight through edge weights, which may not reflect true importance. Although topological features like persistence diagrams and Betti numbers are mathematically clear, connecting them to pharmacological mechanisms remains an open interdisciplinary challenge.

5.4 Scalability of Topological Methods

While dimension-0 and dimension-1 persistent homology can be computed in (O(m \log m)), higher-dimensional cases scale poorly ((O(m^d))). For large graphs like DRKG, even repeated lower-dimensional computations are expensive. Mini-batch training is also challenging since persistent homology is a global property and cannot be applied to subgraphs without loss. Although approximations improve efficiency, they introduce errors, making scalable and accurate implementations an ongoing challenge.

5.5 Generalizability Across Disease Areas

GNN models for drug repurposing are typically trained on single disease areas or balanced datasets, with limited cross-disease evaluation. This makes predicting for rare or emerging diseases difficult due to shifts in graph structure (e.g., cancer vs. neurological networks). The challenge is greater for topology-aware methods, as learned filtrations may not transfer well across different disease subgraphs.

VI. FUTURE

VII. RESEARCH DIRECTIONS

6.1 Explainable Topological GNNs

Integrating explainability with topological GNNs is crucial for clinical use. Future work should use gradient-based methods (e.g., Grad-CAM, Integrated Gradients) to identify key topological features driving predictions. Linking these to specific drug substructures and disease pathways would enable pharmacologists to prioritize candidates based on biological plausibility, not just model scores.

6.2 Learnable Multi-Filtration Strategies for Molecular Graphs

Horn et al. (2022) showed that learnable filtrations outperform fixed ones on complex datasets. Future work should explore domain-specific filtrations for drug graphs, such as electronegativity, bond order, and pharmacophore features, to incorporate chemical knowledge. Multi-filtration approaches that capture multiple molecular properties could further improve topological representations for drug–disease prediction.

6.3 Multi-Omics Integration via Heterogeneous Topological GNNs

Current drug repurposing graphs mainly use transcriptomic and proteomic data. Integrating multi-omics data (genomics, epigenomics, metabolomics, and single-cell RNA sequencing) can better capture patient heterogeneity. Heterogeneous topological GNNs with persistent homology over multi-layer graphs provide a principled way to combine these data types, enabling precision medicine and patient-specific drug repurposing.

6.4 Federated Topological GNNs for Privacy-Preserving Repurposing

Hospital EHR data is valuable for drug response but cannot be centralized due to privacy laws (HIPAA, GDPR). Federated learning enables local GNN training with shared updates, but persistent homology is hard to compute in a distributed setting. Preserving topology while ensuring privacy remains an open challenge.

CONCLUSION

Drug repurposing is a key AI application in biomedicine for accelerating therapy discovery. This survey examines topological GNNs using graph theory and algebraic topology, showing that standard GNNs are limited by the Weisfeiler-Lehman bound and struggle with cyclic structures, while topology-aware models like TOGL use persistent homology for better expressivity. A four-class taxonomy shows topology-aware methods perform best in drug–disease prediction, though challenges remain in scalability, bias, interpretability, and generalization. Future directions include explainable GNNs, multi-omics integration, and federated learning.

REFERENCES

[1] M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt, "Topological graph neural networks," in Proc. 10th Int. Conf. Learn. Representations (ICLR), 2022.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. 5th Int. Conf. Learn. Representations (ICLR), 2017.

[3] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in Proc. 7th Int. Conf. Learn. Representations (ICLR), 2019.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in Proc. Int. Conf. Learn. Representations (ICLR), 2018.

[5] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[6] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in Proc. ESWC, 2018.

[7] D. S. Himmelstein et al., "Systematic integration of biomedical knowledge prioritizes drugs for repurposing," eLife, vol. 6, 2017.

[8] S. V. Ioannidis et al., "Drug repurposing using link prediction on knowledge graphs," arXiv preprint arXiv:2001.00138, 2020.