BG-δ-k-WLNM: Drug-Target Interaction Prediction Framework Using a Bipartite Graph Model and Local-k-dimensional Weisfeiler-Lehman Neural Machine

Maryam Fayaz Roohi, Yangjun Chen
Department of Computer Science, The University of Winnipeg, Winnipeg, Canada

ABSTRACT

The Drug–Target Interaction (DTI) prediction is a crucial task in drug discovery and drug repositioning. It is challenging due to the limited number of known drug–target pairs. Predicting missing drug-target relationships can help speed up the process of identifying unknown interactions between chemical drugs and target proteins in pharmaceutical research. In this paper, we propose a new framework named *BG-δ-k-WLNM* for drug-target interaction prediction that learns latent features from the drug-target interaction local network. We show that our approach is able to learn sophisticated drug-target topological features and outperforms other similarity-based methods in terms of AUROC.

KEYWORDS

DTI prediction, bipartite graphs, deep learning, Weisfeiler-Lehman, Link Prediction, Neural Network, Drug discovery.

I. INTRODUCTION

The computational prediction of drug-target interactions (DTIs) significantly enhances drug discovery, reducing the high costs and long timelines associated with traditional methods. In 2024, the U.S. Food and Drug Administration (FDA) approved 50 new drugs [1]. As the clinical trials are expensive and the experimental approaches for potential drug-target interactions remain challenging, computational prediction methods are needed to address this issue. Computational prediction of DTIs provides a scalable alternative, enabling the early filtering of candidate interactions to prioritize for lab experiments, thus significantly reducing both financial and temporal costs [2]. A rich spectrum of computational strategies has been explored. Ligand-based methods [4] infer interactions from molecular similarity, whereas structure-based docking techniques [5] [6] estimate binding affinity using three-dimensional protein-ligand configurations [3]. Meanwhile, machine learning and deep learning models [7], [8] have gained prominence, reframing DTI prediction as a link prediction or classification task. These approaches leverage network topology, molecular representations, and sequence-based features to learn complex interaction patterns from annotated data [9]. Despite these advances, many existing methods are limited by reliance on handcrafted features, shallow heuristics, or abundant labeled data. Similarity-based and kernel methods capture only local or first-order interactions, while deep learning models may struggle to generalize when training data are scarce or incomplete [9], [10], [11]. To address these limitations, we introduce *BG-δ-k-WLNM*, a novel framework grounded in bipartite graph modeling and local δ–*k*-dimensional Weisfeiler–Lehman neural machinery for DTI prediction. Our method builds a semi-bipartite graph combining known drug–target interactions with drug–drug and protein–protein similarity links. We generate positive samples from known interactions and infer likely negative samples from unlabeled pairs. From each candidate drug–target pair, we pool *k-tuples* and apply δ-*k* Weisfeiler–Lehman labeling to consistently order nodes, a crucial step for preserving structural motifs across subgraphs. These ordered subgraphs are then embedded and processed by deep neural networks to learn rich, nonlinear topological features. Finally, prediction performance is evaluated via *AUROC* against both similarity-based heuristics and graph-based baselines.

II. METHODS

A. Similarity indices

Similarity indices are heuristics used for link prediction, as shown in the **Table. 1**. They can be categorized into: first-order, second-order, and higher-order heuristic methods, based on the most distant node necessary for computing the heuristic [12]. **Table. 2**, shows the *WL*-based methods for link prediction. The similarity indices used in this section, for the purpose of comparison to our approach, are listed as follows:

1. Common Neighbours (*CN*): This measure evaluates the likelihood of a connection between two nodes x and y by counting how many direct neighbors they share. Formally, it is defined as $S_{xy}^{CN} = \Gamma^1(x) \cap \Gamma^1(y)$, where $\Gamma^i(x)$ represents the set of nodes within *i*-hop from x. A larger overlap in their 1-hop neighborhoods indicates a greater chance of a link between x and y.

2. Adamic-Adar (*AA*): This index estimates the similarity between two nodes x and y by considering their shared neighbors, while giving less weight to neighbors that are very common in the network. Specifically, each shared neighbor contributes inversely to the logarithm of its degree, which reduces the influence of highly connected nodes [13]. Mathematically, the score is expressed as

$$S\,(x, y) = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1}{log\,|\Gamma_z|}$$

3. Jaccard: The Jaccard measure evaluates the likelihood that two nodes x and y share a common feature by comparing the overlap of their neighborhoods relative to the union of those neighborhoods [14]. In essence, it can be viewed as a normalized variant of the Common Neighbors (*CN*) index, since it adjusts for the overall size of each node's neighbor set and accounts for their relative influence in the network. The formula is given as

$$S_{xy}^{\text{Jaccard}} = \frac{\Gamma^1(x) \cap \Gamma^1(y)}{\Gamma^1(x) \cap \Gamma^1(y)}$$

4. Resource Allocation (*RA*): The *RA* index measures the similarity between two nodes x and y by simulating the distribution of resources through their shared neighbors. In this model, each common neighbor z of x and y receives one unit of resource from x and then distributes it evenly among all of its neighbors[15]. The total amount of resource that reaches y reflects the strength of the similarity between the two nodes. Formally, the score is defined as

$$S(x, y) = \sum_{z\in \Gamma_x \cap \Gamma_y} \frac{1}{|\Gamma_z|}$$

5. Preferential Attachment (*PA*): This metric is grounded in the principle that nodes with higher connectivity are more likely to form new links. In other words, the probability of a link emerging between two nodes x and y depends directly on their degrees, regardless of whether they share neighbors. Thus, nodes with many connections tend to attract even more [16]. The *PA* score is computed as

$$S_{PA}\ (x,y) = |\Gamma_x| \cdot |\Gamma_y|$$

The *CN*, Jaccard, and *PA* indices are categorized as first-order heuristics [15] because they only rely on the immediate neighbors of the two nodes under consideration. However, in drug–target interaction (DTI) networks, which are inherently bipartite, the neighbors of drug nodes are exclusively proteins, and the neighbors of protein nodes are only drugs. As a result, drugs and targets do not share common neighbors, making the values of *CN* and Jaccard equal to zero. To address this, we apply modified versions of *CN* and Jaccard, defined as:

$$S_{xy}^{CN} = \Gamma^1(x) \cap \Gamma^2(y)$$
$$S_{xy}^{\text{Jaccard}} = \frac{\Gamma^1\,(\mathrm{x}) \cap \Gamma^2(\mathrm{y})}{\Gamma^1\,(\mathrm{x}) \cap \Gamma^2(\mathrm{y})}$$

These modifications turn *CN* and the Jaccard index into second-order heuristics. Unlike these, *PA* only depends on the degree of the nodes x and y and, therefore, can be applied to a bipartite graph.

6. Katz: The Katz measurement evaluates the similarity between two nodes by counting all possible paths that connect them. To avoid overemphasizing long paths, each path of length l is weighted by a damping factor β^l , where β is a constant attenuation parameter ($0<\beta<1/\ \lambda_{max}$). This ensures that shorter paths contribute more to the similarity score, while longer paths are progressively down-weighted. Because it considers paths of arbitrary length across the entire graph, Katz is classified as a high-order heuristic. The formulation is:

$$S_{xy}^{Katz} = \sum_{l=1}^{\infty} \beta^l\,(A^l)_{xy}$$

7. Random Walk (*RW*): This similarity index estimates the likelihood of a connection between two nodes by simulating a stochastic process that moves through the network. Starting from a given node, a walker randomly transitions to one of its neighbors at each step, according to predefined probabilities. The probability that a walker starting from node x eventually reaches node y serves as the similarity score between the two nodes. Unlike local indices such as *CN* or Jaccard, *RW* exploits path information of varying lengths, thereby capturing both local and global structural patterns within the network [17].

$$S_{xy}^{RW} = \sum_{l=1}^{\infty} \alpha^l\,(P^l)_{xy}$$

8. Weisfeiler–Lehman Neural Machine (*WLNM*): This is a framework for link prediction that combines local subgraph extraction, *Weisfeiler–Lehman* (*WL*)–based graph labeling, and neural networks. The goal is to automatically learn structural patterns that indicate whether a link should exist, instead of relying on fixed heuristics. For each candidate link, it extracts a fixed-size enclosing subgraph that captures its local structure. The subgraph is then encoded into an adjacency matrix using a custom graph labeling method called Classic-WL, which orders vertices based on their distance to the link and their structural roles. The resulting matrices are fed into a neural network to learn nonlinear patterns indicative of link existence [18].

9. Bipartite-Graph-Weisfeiler–Lehman Neural Machine (*BG-WLNM*): This framework predicts drug-target interactions using deep learning. It contains enclosing subgraphs that are then encoded into adjacency matrices. A graph labeling algorithm, *Pallete-WL*, is used to impose a consistent node ordering based on geometric proximity and structural role within the graph [19].

Algorithms	**Order**	**Approach**	**Input Req**	**Equation**	**References**
Common Neighbors (*CN*)	First	Local	1-hop neighbors	$S_{xy}^{CN} = \Gamma^1(x) \cap \Gamma^1(y)$	[20]
Preferential Attachment (*PA*)	First	Local	Degree of nodes	$sPA\ (x,y) = \Gamma_x \cdot \Gamma_y$	[16]
Jaccard Index	first	Local	1-hop neighbors	$S_{xy}^{Jaccard} = \frac{\Gamma^1(x) \cap \Gamma^1(y)}{\Gamma^1(x) \cap \Gamma^1(y)}$	[14]
Adamic-Adar (*AA*)	Second	Local	1-hop neighbors	$S\ (x,\ y) = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1}{log\ \lvert\Gamma_z\rvert}$	[13]
Resource Allocation (*RA*)	Second	Local	1-hop neighbors	$S\ (x,\ y) = \sum_{z \in \Gamma_x \cap \Gamma_y} \frac{1}{\lvert\Gamma_Z\rvert}$	[15]
Katz	High	Global	All paths	$S_{xy}^{Katz} = \sum_{l=1}^{\infty} \beta^l\ (A^l)_{xy}$	[21]
Random Walk (*RW*)	High	Global	Transition matrix	$S_{xy}^{CN} = \sum_{l=1}^{\infty} \alpha^l\ (P^l)_{xy}$	[17]

Table. 1 Popular Heuristic for link prediction

Algorithms	Order	Approach	Graph labeling method	Input Req
Weisfeiler-Lehman neural machine (*WLNM*)	High	Local	Classic *WL*	1-hop neighbors
Bipartite-graph-Weisfeiler-Lehman neural machine (*BG-WLNM*)	High	Local	Palette-*WL*	1-hop neighbors
Bipartite-graph-local-δ-k-dimensional-Weisfeiler-Lehman neural machine (*BG-δ-K-WLNM*)	High	Local	Local δ-*K-LWL*	1-hop neighbors

Table. 2 Deep learning-based methods for link prediction

III. PREDICTION METHODOLOGY

A. DTI problem formulation

To capture the topological structure of drug-target relationships, we represent the interaction network as a semi-bipartite graph defined as:

$$G =< D, T, E, F, H >$$

Where D and T denote the sets of drug and target (protein) nodes, respectively. The set $E \subset D \times T$ contains observed drug-target interactions, while $F \subset D \times D$ and $H \subset T \times T$ represent similarities among drugs and targets, respectively. This structure facilitates the integration of both inter-class interactions and intra-class similarities into a unified framework. The parameter k defines the size of the node tuples used to explore the graph structure. We predefine $k = 2$, meaning the algorithm operates over 2-tuples of nodes. This choice is both intuitive and optimal for DTI prediction, as each 2-tuple directly corresponds to a candidate drug-target pair, enabling the model to align naturally with the binary link prediction task. Furthermore, the inclusion of drug-drug and target-target similarities ensures that first-order neighborhoods provide sufficient biological context for each pair, making 2-tuples expressive enough to capture relevant structural and relational patterns. This configuration strikes a balance between model expressiveness and computational feasibility, especially given the size and sparsity typical of real-world DTI graphs. While higher-order settings (with $k > 2$) theoretically offer access to more complex interactions, they introduce significant computational overhead with complexity scaling. Moreover, in sparse interaction graphs, higher-order tuples are more prone to overfitting and may encode redundant or noisy relationships. As a result, in this work, we utilize 2-tuples as the core structural unit for neighborhood-based refinement algorithms. Formally, a 2-tuple is defined as an ordered pair of nodes from the combined set of drugs and targets:

$$(v_1, (v_2) \in (D \cup T)^2$$

This definition yields four possible categories of 2-tuples: (*i*) drug–drug (d_i, d_j), (*ii*) target–target (t_i, t_j), (*iii*) drug–target (d_i, t_j), and (*iv*) target–drug (t_i, d_j). While drug–target tuples constitute the primary candidates for interaction prediction, all 2-tuple types are preserved to support the learning of high-order patterns through the incorporation of topological and similarity-based context. Each 2-tuple is treated as a potential link in the graph and serves as a unit of analysis for downstream refinement. Even if a specific tuple does not correspond to a known edge in E, F, or H, it may encode meaningful structural information based on its neighborhood and is therefore retained in the learning process. This design enables the algorithm to leverage both direct associations and the broader graph topology in distinguishing interacting from non-interacting pairs.

B. **Data formatting:**

A key challenge in training models on drug–target interaction (DTI) networks is the scarcity of known interactions (positive samples). For most drug–target pairs, the interaction status remains unknown, corresponding to missing edges in the network. Therefore, most approaches often address this by selecting negative samples randomly [22], [22], [23]from the unobserved pairs, but such random sampling can introduce noise and distort the classifier's decision boundary. A study by Liu et. al. [24] introduces a strategy to identify reliable negative samples to overcome this challenge. The underlying assumption is that a drug with little or no similarity to any of the known binders of a target is unlikely to interact with that target, and the same holds in the reverse case. To construct this set, we first generate a pool of candidate drug–target pairs, excluding all known positives. Each candidate is represented as a triplet (d_i, t_j, S_{ij}) where S_{ij} denotes the similarity-based score between the drug d_i and target t_j. This score is calculated in two parts:

$$S_{ij}^{DT} = \sum_{T_k \in T_{D_i}} S_{T_j T_k}^{T}$$

which aggregates the similarity of the target t_j With all targets already interacting with the drug d_i; and

$$S_{ji}^{TD} = \sum_{D_k \in D_{T_j}} S_{D_j D_k}^{D}$$

which aggregates the similarity of the drug d_i with all drugs known to interact with the target t_j. The final similarity measure is then defined as

$$S_{ij} = e^{-(S_{ij}^{DT} + S_{ji}^{TD})}$$

Candidate pairs are ranked in descending order of this score, and those with the highest values are selected as reliable negative samples. These negatives, combined with randomly sampled positives from known interactions, form the training set for the neural network classifier.

C. **Dataset**

For this study, we employed a publicly available benchmark dataset encompassing four major categories of protein targets: nuclear receptors (*NR*), G protein-coupled receptors (*GPCR*), ion channels (*IC*), and enzymes (*E*) [25]. The dataset statistics, including the number of drugs, number of targets, observed interactions, maximum possible interactions, and calculated sparsity, are summarized in the accompanying table. Sparsity is defined as:

$$1 - \frac{Number\ of\ known\ interactions}{Max\ possible\ intreaction}$$

A sparsity value approaching 1 (or 100%) indicates that only a small fraction of possible interactions is known, which increases the difficulty of prediction due to limited training information. Conversely, smaller sparsity values imply denser interaction networks, providing richer structural information for learning.

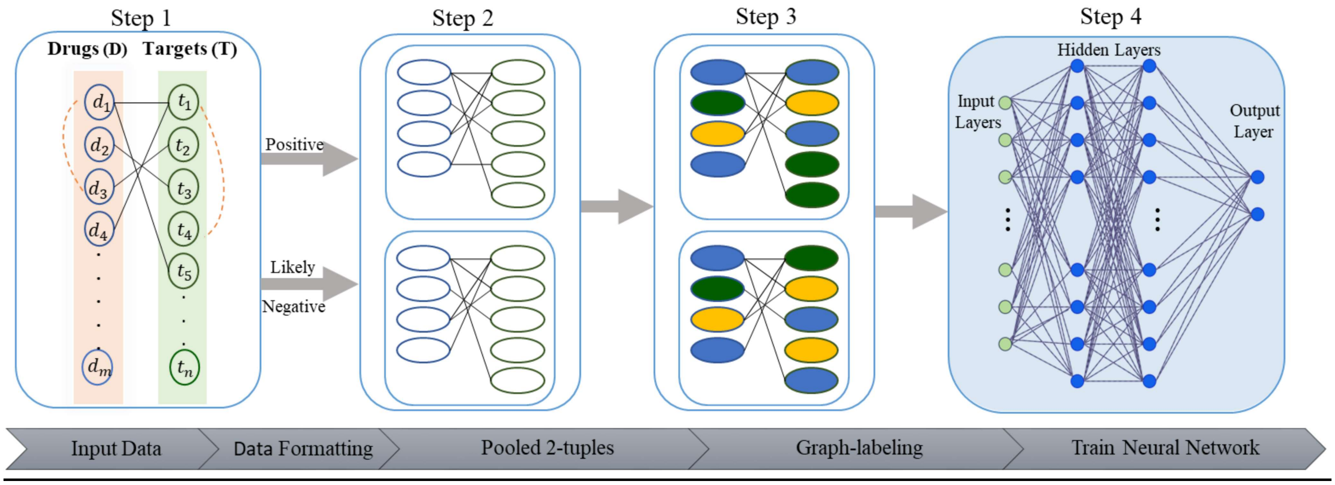


Figure. 1 Model workflow

D. Workflow

The local δ-k-dimensional *Weisfeiler–Leman* algorithm [26], abbreviated *δ-k-LWL*, is a refinement [27], [28] procedure that generalizes the classical *Weisfeiler–Leman* (*WL*) method for graph isomorphism and graph learning, but modifies it in two important ways: it works with *k*-tuples of vertices instead of single vertices, and it restricts the neighborhood aggregation process to local neighborhoods only. The model begins by constructing a semi-bipartite graph combining drug-target interactions and intra-type similarities. For each drug-target pair, all possible *k*-tuples of fixed size that the drug d_i or target t_j is included in a pool. *Local δ-k-WL* is applied to the k-tuples, each of which is labeled through iterative neighborhood-aware refinement. Unlike standard *WL*, *local-δ-k-WL* constrains updates to the k-tuple's local region, improving efficiency and preserving structural specificity. The refinement process used in Palette-WL involves repeated hashing and re-labeling steps to preserve structural consistency. For subgraphs with high-degree nodes, this process becomes computationally intensive, particularly when applied at scale across many subgraphs. The traditional *WL* suffers from high computational cost due to dense, global aggregation. In contrast, *local-δ-k-WL* restricts aggregation to the local neighborhood of each node in the k-tuple. Once the graph labeling is done, the k-tuple chosen is made into an adjacency matrix. Then, as the adjacency matrix is symmetric, the upper triangular matrix is taken and it is fed vertically to a fully connected neural network. As shown in Fig. 1, the proposed link prediction algorithm consists of four steps:

1. **Pool *k*-Tuples:** To capture the complete local structural context surrounding each candidate drug–target pair, we generate all possible ordered 2-tuples of nodes that include either the candidate drug d_i or the candidate target t_j. In the bipartite/semi-

bipartite DTI network, this naturally yields three categories of tuples: (1) Drug–Target tuples *(d, t)*, which represent potential or known interactions. (2) Drug–Drug tuples *(d, d)*, derived from drug–drug similarity or co-neighborhood relations. (3) Target–Target tuples *(t, t)*, derived from target similarity or shared neighborhoods. All three tuple types are pooled because the *local δ-2-WL* refinement algorithm recolors a tuple based on the multiset of colors of its local neighbors, which are obtained by replacing one component of the tuple with an adjacent vertex in the graph. If we restrict pooling only to drug–target pairs, much of this neighborhood structure would be truncated. By including *(d, d)* and *(t, t)* tuples alongside *(d, t)*, the refinement step propagates similarity information within drug and target spaces, enabling the algorithm to distinguish subtler structural motifs and learn higher-order relationships. The total number of possible 2-tuples in the system is given by the binomial coefficient $\binom{|v|}{2} \times 2!$.

```
Input: Semi-bipartite graph $G = (D, T, E, F, H)$, Tuple size $K$, $(d_i, t_j)$
Output: Set of tuples $P(d_i, t_j)$
$P(d_i, t_j) \leftarrow \{(d_i, t_j)\}$
$N_D(d_i) \leftarrow \{d_m \in D \mid (d_i, d_m) \in F$ or $(d_i, t_x) \in E$ for some $t_x\}$
$N_T(t_j) \leftarrow \{t_n \in T \mid (t_j, t_n) \in H$ or $(t_j, d_y) \in E$ for some $d_y\}$
For every $d_m \in N_D(d_i)$: add $(d_m, t_j)$ to $P(d_i, t_j)$
For every $t_n \in N_T(t_j)$: add $(d_i, t_n)$ to $P(d_i, t_j)$
For every $d_m \in N_D(d_i)$: add $(d_i, d_m)$ to $P(d_i, t_j)$
For every $t_n \in N_T(t_j)$: add $(t_j, t_n)$ to $P(d_i, t_j)$
```

Table 3. Pooling 2-Tuples Around a Candidate Pair (di, tj)

2. **Pattern Encoding**: Once *k*-tuples are generated around (d_i, t_j), we apply *local-δ-k-WL* color refinement to assign labels. The refinement algorithm is employed to impose an ordering on pooled *k*-tuples by iteratively updating their labels based on structural context. For a *k*-tuple $V = (V_1, \ldots, V_K)$, initially, each *k*-tuple is assigned a color $C^0(v)$ reflecting its isomorphism type, ensuring that tuples with similar structural roles receive the same label. The algorithm then proceeds iteratively: in each round, the color of a tuple is updated by collecting the colors of its so-called j-neighbors. A *j*-neighbor of a tuple $V = (V_1, \ldots, V_K)$, is obtained by replacing the *j-th* component V_j with every possible vertex *w* of the graph, producing a new tuple $\varphi_j(v, w)$. When constructing a *j*-neighbor $\varphi_j(v, w)$, one checks whether the new vertex *w* is adjacent to the original vertex V_j in the underlying graph. If it is, the new tuple is called a local *j*-neighbor; otherwise, it is a global *j*-neighbor. The *local δ-k-WL* discards the global neighbors entirely and restricts attention only to local neighborhoods. The refinement step constructs, for each position *j*, the multiset of colors of tuples $\varphi_j(v, w)$ where *w* ranges over the adjacency set $N(V_j)$ of the vertex V_j. These multisets, one for each component, are combined with the old color of the tuple to form its new color. The neighborhood aggregation at iteration *i* computes:

$$M_i^{\delta}(v)=(\{C^i(\varphi_1(v,w))/w\in N(V_1)\},\ldots,\{C^i(\varphi_k(v,w))/w\in N(V_k)\})$$

where C^i is the current coloring function after i rounds of refinement. For each tuple u, C^i(u) is the label or "color" assigned at round i. $N(V_j)$ is the set of neighbors of the vertex V_j in the graph. This restriction to true neighbors is what makes it local. So $M_i^{\delta}(v)$ encodes the local structural context of tuple v at iteration i, by recording how its color relates to the colors of all its local neighbor tuples. At each iteration, *δ-k-LWL* recolors every k-tuple by combining its current color with the "fingerprint" of its neighborhood, where the neighborhood is defined as all k-tuples you get by swapping out one vertex in the tuple with an actual graph neighbor. The aggregation formula gathers this fingerprint. The refinement update formula specifies the new color of v at round $i + 1$.

$$C^{i+1}(v)=(C^i(v),M_i^{\delta}(v))$$

In practice, this pair is turned into a unique integer via a deterministic scheme (e.g., lexicographic serialization of the k multisets followed by hashing or stable renaming). Two tuples u and v therefore get the same new color at round $i + 1$ if they had the same color at round i and their per-position neighborhood multisets match exactly.

The process is iterated in successive rounds until convergence is reached, producing a stable partition of the k-tuples. The effect is that each tuple's identity becomes tied to the structure of its immediate neighborhood, with information propagated outward over multiple refinement rounds. The final labels serve as a canonical representation of the tuples, making them comparable across tuples. Such encoding captures higher-order structural features essential for link prediction tasks. To ground this in a concrete $k = 2$ setting, (e.g., a drug-target bipartite graph): a tuple $V = (d, t)$ initially receives a color. At round i, two multisets are formed. The first collects the colors of $\varphi_1(v,w) = (w ,t)$ for all drugs w that interact with t (neighbors of t); the second collects the colors of $\varphi_2(v,\ w') = (d,\ w')$ for all targets w' that interact with d (neighbors of d).These two multisets, concatenated in order, are the fingerprint $M_i^{\delta}(d,\ t)$. These fingerprints are paired with the current color $C^i(d,\ t)$ and compressed to the next color $C^{i+1}(d,\ t)$. Over successive rounds, tuples sitting in locally different motifs (e.g., high-degree drug with low-degree target versus the reverse; different second-shell neighborhoods when it is iterated) quickly diverge to different colors, while tuples in isomorphic local neighborhoods converge to the same color class. Because only true adjacencies are explored when building the fingerprint, the computation respects graph sparsity and empirically overfits less than global *k-WL* (colors grow more slowly).

3. **Aggregate Tuple Features**: After labeling all k-tuples, we transform these labels into a numerical feature representation suitable for input to a neural network. Each k-tuple is first assigned a label through the refinement process, which encodes its structural role in the graph. To make these categorical labels compatible with a machine learning model, we map them to a numeric form using integer encoding: every unique color is

assigned a distinct integer index, ensuring a one-to-one correspondence between structural labels and numeric identifiers. These integer-encoded color IDs are then passed to an embedding layer within the neural network. The embedding layer maps each color ID to a dense, fixed-size vector, which in our implementation is set to 16 dimensions. This dimensionality does not represent explicit features; rather, it defines the size of the continuous vector space in which the model will learn to represent structural patterns. Initially, these embedding vectors are initialized, and their values are optimized during training through backpropagation. Consequently, the embedding process enables the network to learn latent structural characteristics of *k*-tuples. By replacing sparse categorical labels with dense, trainable vectors, this approach produces compact, information-rich representations that are well-suited for fully connected neural networks. It avoids the limitations and the redundancy of adjacency-based encoding for small tuples, while remaining computationally efficient and expressive enough to model higher-order structural relationships.

4. **Learning phase by neural network**: After encoding each tuple into a fixed-length feature vector derived from its labeled *k*-tuples, these vectors are provided as input to a fully connected neural network. During training, the network learns to capture nonlinear topological patterns associated with interacting and non-interacting drug–target pairs. Once trained, the model can be applied to predict interactions for unseen drug–target pairs. The neural network outputs a probability estimate, predicting the interaction between testing drug-target pair.

IV. EXPERIMENTAL RESULTS

A. Evaluation Metrics

A 10-fold cross-validation protocol was employed to evaluate the predictive performance of the proposed framework. The dataset was partitioned into ten mutually exclusive subsets, with nine subsets used for model training and the remaining subset reserved for testing in each fold. A neural network architecture was used with three fully-connected layers with 32, 32, and 16 neurons, respectively, each employing the Rectified Linear Unit (*ReLU*) activation function. The output layer consisted of a Softmax classifier, which produced the predicted probability of interaction for each candidate pair. Model performance was quantified using the Area Under the Receiver Operating Characteristic Curve (*AUROC*), which evaluates the trade-off between the true positive rate (*TPR*) and the false positive rate (*FPR*):

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

where TP, T, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. AUROC was selected due to its robustness to class imbalance and its widespread adoption as a standard metric in drug–target interaction prediction.

B. Dataset

For this study, we employed a publicly available benchmark dataset encompassing four major categories of protein targets: nuclear receptors (*NR*), *G* protein-coupled receptors (*GPCR*), ion channels (*IC*), and enzymes (*E*) [25]. The dataset statistics, including the number of drugs, number of targets, observed interactions, maximum possible interactions, and calculated sparsity, are summarized in the accompanying table. Sparsity is defined as:

$$Sparsity = 1 - \left(\frac{Number\ of\ known\ interactions}{Max\ possible\ intreaction} \right)$$

A sparsity value approaching 1 (or 100%) indicates that only a small fraction of possible interactions is known, which increases the difficulty of prediction due to limited training information. Conversely, smaller sparsity values imply denser interaction networks, providing richer structural information for learning.

Statistics	**Drugs**	**Targets**	**Known Interactions**	**Max possible interactions**	**Sparsity**
Nuclear receptor (*NR*)	54	26	90	1,404	93.5%
GPCR	223	95	635	21,185	97%
Ion channel (*IC*)	210	204	1476	4,284	96.5%
Enzyme	445	664	2926	295,480	99%

Table 4: Dataset Specification

C. Results

We comprehensively compared our approach with heuristic similarity-based methods[29] for drug-target interaction predictions reported in the literature, namely the performance of our method (*BG-δ-k-WLNM*) compared with *CN*, Jaccard Index, Katz Index, Adamic-Adar (*AA*), Random Walk (*RA*), and PA in terms of *AUROC* on *NR*, *GPCR*, *IC*, and E datasets. In addition, we also compared our model with deep-learning based methods namely Weisfeiler-Lehman neural machine (*WLNM*) and Bipartite-graph-Weisfeiler-Lehman neural machine (*BG-WLNM*).

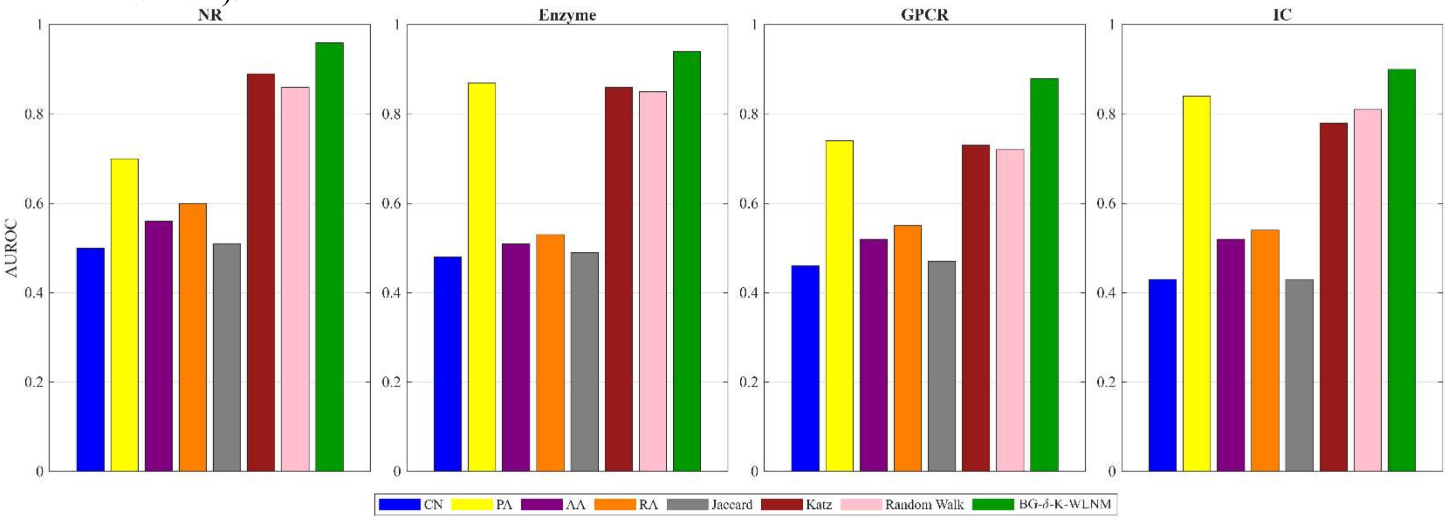


Figure. 2: Comparing the *AUROC* performance metric of our method (*BG-δ-K-WLNM*) with others for four datasets.

All methods exhibited relatively lower performance on the NR dataset, primarily due to insufficient positive training samples (Fig. 2). In this dataset, *CN* achieved nearly identical results across all cases, as it relies solely on the common-neighbors metric. In contrast, higher-order heuristics, such as Katz and *RW*, performed better by leveraging longer paths. For the Enzyme dataset, the graphs were highly sparse, resulting in minimal neighbor overlap. **Figure 3** presents the results of a 10-fold cross-validation comparison between our proposed *BG-δ-k-WLNM* and two *WL*-based baselines (*WLNM* and *BGWLNM*) across four benchmark datasets (*NR*, *Enzyme*, *GPCR*, and *IC*). The evaluation was performed under three sampling conditions: balanced (α = 50%), moderately imbalanced (α = 10%), and extremely imbalanced (*α = 0.11%*). Under balanced conditions, BG-δ-k-WLNM consistently outperformed the baseline methods, achieving *AUROC* improvements of 18.5% in the *NR* dataset, 4.6% in the Enzyme dataset, 6.3% in the *GPCR* dataset, and 4.6% in the IC dataset. These results demonstrate that our model effectively captures discriminative structural features when at least one of the drug or target nodes has known interactions. In contrast, performance declined when both the drug d_i and target t_j were novel. This limitation could potentially be mitigated by incorporating auxiliary information, such as drug–drug and protein–protein similarity networks, to enrich the representation of novel pairs. The robustness of *BG-δ-k-WLNM* is further highlighted under imbalanced scenarios, which better reflect the sparsity of real-world DTI networks. When negative pairs were sampled at ten times the number of positives (α = 10%), *AUROC* values decreased for all methods; however, our model consistently maintained superior performance over the baselines. Even in the most challenging case, where all unknown interactions were considered as negatives (α = 0.11%), *BG-δ-k-WLNM* continued to outperform other *WL*-based approaches. Overall, these results confirm that *BG-δ-k-WLNM* successfully learns higher-order structural patterns beyond immediate neighbors by systematically encoding local tuple structures. Its adaptive graph labeling ensures that nodes with similar topological roles receive consistent representations, thereby enhancing discriminative power and achieving robust predictive performance across varying levels of class imbalance.

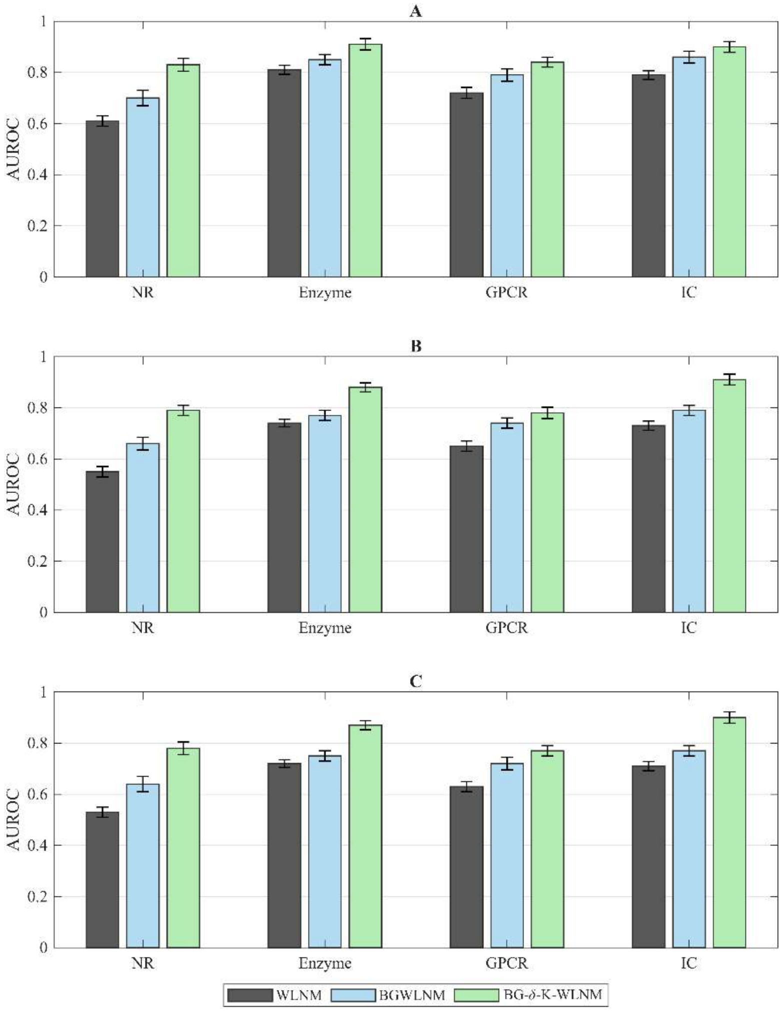
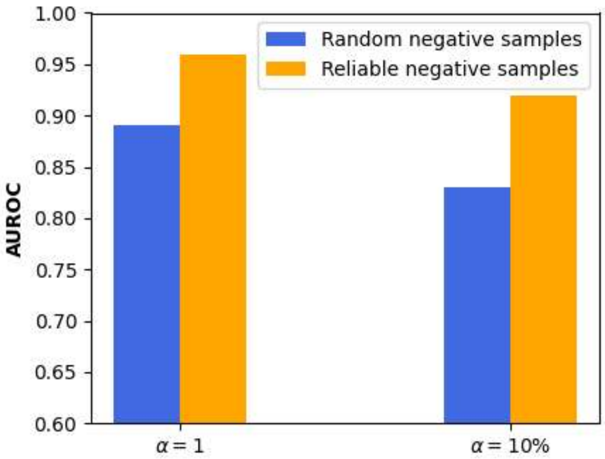


Figure 3. 10-fold cross-validation performance evaluation of our approach compared with the *WL*-based method in terms of *AUROC*. **A:** *AUROC* scores in which all methods are trained and tested on balanced datasets, **B**: AUROC scores in which the number of negative samples was 10 times more than the number of positive samples (α = 10%), C: *AUROC* scores in which all unknown drug-target interacting pairs are considered (α = 0.11%). All results were summarized over 10 trials and expressed as mean±SD.

To evaluate the impact of negative sampling on model performance, we conducted experiments using the Ion Channel (*IC*) dataset. Among the four benchmark datasets, *IC* offers the most balanced trade-off between size and complexity: it contains a sufficiently large number of known interactions (1,476) to ensure robust statistical evaluation, while remaining computationally manageable compared to the much larger Enzyme dataset. This makes *IC* particularly suitable for assessing the effect of sampling strategies. We compared two approaches: random sampling of unknown drug–target pairs and the reliable negative sampling technique described in Subsection "Data formatting." The *AUROC* results are presented in **Figure 4**. As expected, training with reliable negatives consistently outperformed random sampling, and the advantage became more pronounced under imbalanced conditions (e.g., when the positive-to-negative ratio α = 10%). This highlights the necessity of carefully selecting negative samples to improve model robustness in sparse drug–target interaction networks.



Towards

Figure 4. Comparison of the performance of our model with random negative samples and reliable negative samples in terms of AUROC

Future works

The present study demonstrates that our framework moves beyond handcrafted features by enabling the neural network to learn structural information directly from the DTI graph. Central to this approach is the representation of local neighborhoods through K-tuples, which effectively capture the topology surrounding candidate drug–target pairs. While simple first-order measures such as Common Neighbors can be derived from 1-hop subgraphs, prior work has shown that higher-order heuristics, including *Katz*, generally yield stronger predictive performance [15]. This observation is consistent with our comparative experiments (**Figure 2**). A natural assumption is that modeling higher-order relationships requires enlarging the hop size h indefinitely. In practice, however, increasing h inflates tuple size (*K*) and leads to prohibitive computational costs. As highlighted by Zhang et al. [7], high-order graph patterns can often be captured within relatively small neighborhoods, mitigating the need for overly large subgraphs. Our method follows the same principle: by employing the *δ-k-WL* refinement, we preserve both structural roles and directional topological information [7] [18], while keeping computations manageable. Despite these strengths, the current framework still has room for improvement. Since the model encodes *k*-tuples through fixed-size vectors, this can result in the loss of potentially useful structural information. Future work could explore integrating graph neural networks (GNN) [30] that naturally handle variable-size subgraphs and directly leverage node and edge features. Another direction is the incorporation of heterogeneous biological networks. Recent studies emphasize the added value of integrating drug–drug, protein–protein, drug–disease, and drug–side effect associations for improving DTI prediction [31] [32] [33]. Extending our pipeline to jointly model these relations may provide a richer biological context and improve prediction robustness. Lastly, while our evaluation confirms strong *AUROC* performance on benchmark datasets, the most decisive test of predictive utility lies in practical drug discovery. A crucial step forward will be to assess whether the framework can successfully re-identify FDA-approved drug–target pairs and uncover drug repositioning opportunities. Although our model can already prioritize high-confidence candidates for follow-up, a systematic validation against comprehensive FDA data and case studies of novel predictions will be essential to demonstrate clinical relevance [34], [35].

Conclusion

We introduced *BG-δ-k-WLNM*, a deep learning framework for drug–target interaction prediction that combines a semi-bipartite graph representation with local *k-dimensional Weisfeiler–Lehman* refinement. By imposing a consistent node ordering and encoding local *K-tuple* structure, the model learns discriminative higher-order patterns directly from topology rather than hand-engineered features. Across the Yamanishi benchmarks under both balanced and imbalanced protocols, *BG-δ-k-WLNM* demonstrated consistently strong *AUROC* performance relative to heuristic similarity indices and *WL*-based baselines. Our analysis also showed that training with reliable negative samples instead of randomly chosen unknown pairs yields more robust classifiers, an effect that is especially clear on the Ion Channel (*IC*) dataset, which offers a practical balance of scale and sparsity for evaluating sampling strategies. Altogether, these results indicate that localized *WL* refinement with tuple-based encoding is an effective and scalable approach for DTI link prediction.

References

[1] Y. Wang, F. Yang, B. Wang, L. Xie, and W. Chen, "New FDA drug approvals for 2024: Synthesis and clinical application," *Eur J Med Chem*, vol. 285, p. 117241, Mar. 2025, doi: 10.1016/J.EJMECH.2025.117241.

[2] Q. Feng, E. Dueva, A. Cherkasov, and M. Ester, "PADME: A Deep Learning-based Framework for Drug-Target Interaction Prediction," Jul. 2018, Accessed: Aug. 26, 2025. [Online]. Available: https://arxiv.org/pdf/1807.09741

[3] A. Ezzat, M. Wu, X. L. Li, and C. K. Kwoh, "Computational prediction of drug-target interactions using chemogenomic approaches: An empirical survey," *Brief Bioinform*, vol. 20, no. 4, pp. 1337–1357, Mar. 2018, doi: 10.1093/BIB/BBY002,.

[4] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug–target interactions using bipartite local models," *Bioinformatics*, vol. 25, no. 18, p. 2397, Sep. 2009, doi: 10.1093/BIOINFORMATICS/BTP433.

[5] K. A. Loving, A. Lin, and A. C. Cheng, "Structure-Based Druggability Assessment of the Mammalian Structural Proteome with Inclusion of Light Protein Flexibility," *PLoS Comput Biol*, vol. 10, no. 7, p. e1003741, 2014, doi: 10.1371/JOURNAL.PCBI.1003741.

[6] G. M. Morris *et al.*, "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility," *J Comput Chem*, vol. 30, no. 16, pp. 2785–2791, Dec. 2009, doi: 10.1002/JCC.21256')).

[7] M. Zhang and Y. Chen, "Link Prediction Based on Graph Neural Networks," *Adv Neural Inf Process Syst*, vol. 2018-December, pp. 5165–5175, Feb. 2018, Accessed: Aug. 23, 2025. [Online]. Available: https://arxiv.org/pdf/1802.09691

[8] N. Zong, H. Kim, V. Ngo, and O. Harismendy, "Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations," *Bioinformatics*, vol. 33, no. 15, pp. 2337–2344, Aug. 2017, doi: 10.1093/BIOINFORMATICS/BTX160.

[9] W. Shi, H. Yang, L. Xie, X. X. Yin, and Y. Zhang, “A review of machine learning-based methods for predicting drug–target interactions,” *Health Inf Sci Syst*, vol. 12, no. 1, Dec. 2024, doi: 10.1007/S13755-024-00287-6,.

[10] B. Liu, S. Wu, J. Wang, X. Deng, and A. Zhou, “HiGraphDTI: Hierarchical Graph Representation Learning for Drug-Target Interaction Prediction,” Apr. 2024, Accessed: Aug. 26, 2025. [Online]. Available: https://arxiv.org/pdf/2404.10561

[11] J. Peng, J. Li, and X. Shang, “A learning-based method for drug-target interaction prediction based on feature representation learning and deep neural network,” *BMC Bioinformatics*, vol. 21, no. 13, pp. 1–13, Sep. 2020, doi: 10.1186/S12859-020-03677-1/TABLES/3.

[12] V. Martínez, F. Berzal, and J. C. Cubero, “A Survey of Link Prediction in Complex Networks,” *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, pp. 1–33, Dec. 2016, doi: 10.1145/3012704.

[13] L. A. Adamic and E. Adar, “Friends and neighbors on the Web,” *Soc Networks*, vol. 25, no. 3, pp. 211–230, Jul. 2003, doi: 10.1016/S0378-8733(03)00009-1.

[14] D. Liben-Nowell and J. Kleinberg, “The Link Prediction Problem for Social Networks *,” 2004, Accessed: Aug. 23, 2025. [Online]. Available: www.arxiv.org.

[15] L. L. Linyuan and T. Zhou, “Link prediction in complex networks: A survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011, doi: 10.1016/J.PHYSA.2010.11.027.

[16] A.-L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science (1979)*, vol. 286, no. 5439, pp. 509–512, Oct. 1999, doi: 10.1126/science.286.5439.509.

[17] F. Fouss, A. Pirotte, J. M. Renders, and M. Saerens, “Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation,” *IEEE Trans Knowl Data Eng*, vol. 19, no. 3, pp. 355–369, Mar. 2007, doi: 10.1109/TKDE.2007.46.

[18] M. Zhang and Y. Chen, “Weisfeiler-Lehman Neural Machine for Link Prediction,” vol. 9, 2017, doi: 10.1145/3097983.3097996.

[19] H. Eslami Manoochehri and M. Nourani, “Drug-target interaction prediction using semi-bipartite graph model and deep learning,” *BMC Bioinformatics*, vol. 21, no. 4, pp. 1–16, Jul. 2020, doi: 10.1186/S12859-020-3518-6/FIGURES/9.

[20] M. E. J. Newman, “Clustering and preferential attachment in growing networks,” *Phys Rev E*, vol. 64, no. 2, p. 025102, Jul. 2001, doi: 10.1103/PhysRevE.64.025102.

[21] M. P. H. Stumpf *et al.*, “Estimating the size of the human interactome,” *Proc Natl Acad Sci U S A*, vol. 105, no. 19, pp. 6959–6964, May 2008, doi: 10.1073/PNAS.0708078105,.

[22] Z. Li *et al.*, “In silico prediction of drug-target interaction networks based on drug chemical structure and protein sequences,” *Sci Rep*, vol. 7, no. 1, Dec. 2017, doi: 10.1038/S41598-017-10724-0,.

[23] Y. Luo *et al.*, “A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information,” *Nat Commun*, vol. 8, no. 1, Dec. 2017, doi: 10.1038/S41467-017-00680-8,.

[24] H. Liu, J. Sun, J. Guan, J. Zheng, and S. Zhou, “Improving compound–protein interaction prediction by building up highly credible negative samples,” *Bioinformatics*, vol. 31, no. 12, pp. i221–i229, Jun. 2015, doi: 10.1093/BIOINFORMATICS/BTV256.

[25] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, “Prediction of drug–target interaction networks from the integration of chemical and genomic spaces,” *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, Jul. 2008, doi: 10.1093/BIOINFORMATICS/BTN162.

[26] C. Morris, G. Rattan, and P. Mutzel, “Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings,” *Adv Neural Inf Process Syst*, vol. 2020-December, Apr. 2019, Accessed: Aug. 30, 2025. [Online]. Available: https://arxiv.org/pdf/1904.01543

[27] B. Y. Weisfeiler and A. A. Leman, “THE REDUCTION OF A GRAPH TO CANONICAL FORM AND THE ALGEBRA WHICH APPEARS THEREIN”.

[28] P. Schweitzer PASCAL *et al.*, “Weisfeiler-Lehman Graph Kernels Nino Shervashidze Kurt Mehlhorn Karsten M. Borgwardt,” *Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.

[29] Y. Lu, Y. Guo, and A. Korhonen, “Link prediction in drug-target interactions network using similarity indices,” *BMC Bioinformatics*, vol. 18, no. 1, pp. 1–9, Jan. 2017, doi: 10.1186/S12859-017-1460-Z/FIGURES/5.

[30] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Trans Neural Netw*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[31] F. Wan, L. Hong, A. Xiao, T. Jiang, and J. Zeng, “NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions,” *Bioinformatics*, vol. 35, no. 1, pp. 104–111, Jan. 2019, doi: 10.1093/BIOINFORMATICS/BTY543.

[32] Y. Luo *et al.*, “A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information,” *Nat Commun*, vol. 8, no. 1, pp. 1–13, Dec. 2017, doi: 10.1038/S41467-017-00680-8;SUBJMETA=114,1305,2248,2401,2408,631;KWRD=DATA+INTEGRATION,MACHINE+LEARNING,NETWORK+TOPOLOGY,VIRTUAL+DRUG+SCREENING.

[33] X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, and F. Cheng, “DeepDR: A network-based deep learning approach to in silico drug repositioning,” *Bioinformatics*, vol. 35, no. 24, pp. 5191–5198, Dec. 2019, doi: 10.1093/BIOINFORMATICS/BTZ418,.

[34] M. Zitnik, M. Agrawal, and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks,” *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, Jul. 2018, doi: 10.1093/BIOINFORMATICS/BTY294.

[35] M. J. Keiser *et al.*, “Predicting new molecular targets for known drugs,” *Nature*, vol. 462, no. 7270, pp. 175–181, Nov. 2009, doi: 10.1038/NATURE08506,.