



GADTI: Graph Autoencoder Approach for DTI Prediction From Heterogeneous Network

Zhixian Liu^{1,2}, Qingfeng Chen^{3*}, Wei Lan³, Haiming Pan³, Xinkun Hao³ and Shirui Pan⁴

¹ School of Medical, Guangxi University, Nanning, China, ² School of Electronics and Information Engineering, Beibu Gulf University, Qinzhou, China, ³ School of Computer, Electronic and Information, Guangxi University, Nanning, China,

⁴ Department of Data Science and AI, Monash University, Melbourne, VIC, Australia

Identifying drug–target interaction (DTI) is the basis for drug development. However, the method of using biochemical experiments to discover drug–target interactions has low coverage and high costs. Many computational methods have been developed to predict potential drug–target interactions based on known drug–target interactions, but the accuracy of these methods still needs to be improved. In this article, a graph autoencoder approach for DTI prediction (GADTI) was proposed to discover potential interactions between drugs and targets using a heterogeneous network, which integrates diverse drug-related and target-related datasets. Its encoder consists of two components: a graph convolutional network (GCN) and a random walk with restart (RWR). And the decoder is DistMult, a matrix factorization model, using embedding vectors from encoder to discover potential DTIs. The combination of GCN and RWR can provide nodes with more information through a larger neighborhood, and it can also avoid over-smoothing and computational complexity caused by multi-layer message passing. Based on the 10-fold cross-validation, we conduct three experiments in different scenarios. The results show that GADTI is superior to the baseline methods in both the area under the receiver operator characteristic curve and the area under the precision–recall curve. In addition, based on the latest Drugbank dataset (V5.1.8), the case study shows that 54.8% of new approved DTIs are predicted by GADTI.

Keywords: drug–target interaction prediction, network embedding, graph convolutional network, autoencoder, random walk, heterogeneous network

INTRODUCTION

The drug acts on the target protein, thereby affecting the expression of the target protein to achieve the therapeutic effect on the disease. Therefore, finding drug–target interactions is the basis of drug development. The research and development of innovative drugs often requires billions of dollars and more than a decade of work, and usually ends in failure. Hence, it is an important choice for pharmaceutical companies to discover potential drug–target interactions (DTIs) by using the known DTIs. The properties of existing drugs are familiar to people, and their safety is guaranteed. However, there are some limits in both coverage and throughput of biochemical experiments to identify new DTIs. Consequently, the prediction of DTIs using computational methods has attracted extensive attention.

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Khanh N. Q. Le,
Taipei Medical University, Taiwan
Yusen Zhang,
Shandong University, China

*Correspondence:

Qingfeng Chen
qingfeng@gxu.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 08 January 2021

Accepted: 12 March 2021

Published: 09 April 2021

Citation:

Liu Z, Chen Q, Lan W, Pan H, Hao X
and Pan S (2021) GADTI: Graph
Autoencoder Approach for DTI
Prediction From Heterogeneous
Network. *Front. Genet.* 12:650821.
doi: 10.3389/fgene.2021.650821

Early computational methods were mainly based on drug–drug similarity and target–target similarity, or the features of drugs and targets. Some of the methods based on similarity first calculate the similarity between the drug pairs (e.g., chemical structure similarity) and the similarity between the target pairs (e.g., protein sequence similarity), and then use the known DTIs to score the unknown DTI (Cheng et al., 2012; Mei et al., 2013; Wang et al., 2014). Other similarity-based methods process a random walk on the network composed of multiple data sources, such as drug–drug interactions, target–target interactions, and DTIs to obtain the similarity between nodes to predict new DTIs (Chen et al., 2012; Seal et al., 2015). In the methods based on features, both the drugs and the targets are represented as fixed-length feature vectors, and the known drug–target pairs are divided into positive and negative categories. Then the DTI prediction is transformed into a binary classification problem. Machine learning methods such as support vector machines, random forests, and conditional random fields can be directly used for prediction (Nagamine et al., 2009; Lan et al., 2016; Olayan et al., 2018; Chen et al., 2019; Shi et al., 2019).

In recent years, network embedding methods (Perozzi et al., 2014) have shown excellent performance in network data analysis (Cai et al., 2017), and have been introduced into DTI prediction (Su et al., 2018; Bagherian et al., 2020; Liu et al., 2020). Network embedding is also known as graph embedding. In network embedding, nodes such as drugs and targets can all be converted into low-dimensional vectors that represent their features and can be directly used for DTI prediction. The main methods of network embedding include matrix factorization, random walk, and deep learning.

A multiple similarities collaborative matrix factorization model (Zheng et al., 2013) was proposed to predict DTI. It incorporates anatomical therapeutic chemical similarity and chemical structure similarity of drugs, as well as genomic sequence similarity, gene ontology (GO) similarity, and protein–protein interaction (PPI) network similarity of targets. A combination of these similarity matrices was used to approximate the drug feature matrix D and the target feature matrix T , and then the inner product between D and T was utilized to approximate the DTI matrix. TriModel (Mohamed et al., 2019) uses the drug-related knowledge graph to find potential DTIs. It learns the feature vectors of nodes in the knowledge graph through tensor decomposition. These vectors are used to determine whether the drug and the target interact. Meanwhile, DTINet (Luo et al., 2017) first uses the random walk to obtain the low-dimensional feature vector of each drug and protein, projects the drug vector and protein vector into the same space, and then discovers new interactions through matrix completion. Encouraged by the DeepWalk (Perozzi et al., 2014) model, some researchers have combined the random walk with shallow neural networks (Zong et al., 2017, 2019; Zhu et al., 2018). These methods first construct a heterogeneous network based on multiple data sources, and then apply DeepWalk, node2vec (Grover and Leskovec, 2016), and other algorithms to the network to obtain the embedding vectors of drug nodes and target nodes. NeoDTI (Wan et al., 2019) uses a deep learning method based on neighborhood information aggregation to

discover new DTIs. It aggregates neighbor information based on edge types in heterogeneous networks. Then, the feature vector of the node is used to reconstruct the original network. There are also several studies based on drug structure and protein sequence (Wen et al., 2017; Karimi et al., 2019; Öztürk et al., 2019). Starting from the chemical structure and protein sequence of compounds, deep learning methods are then employed to predict drug–target binding affinity.

Matrix factorization methods can capture the global structure of the network, but its space complexity increases rapidly as the network scale increases. Random walk methods are more efficient because they usually gather only local features. Deep learning methods are outstanding in DTI prediction because it can discover hidden features and associations from multi-source heterogeneous network, and it is easy to integrate externally associated data of drugs and targets (e.g., GO) to improve performance. However, deep learning is computationally expensive and time-consuming. Among the deep learning methods, the graph convolutional network (GCN)-based message passing (also known as neighborhood information aggregation) algorithms have recently attracted special attention due to their flexibility and good performance (Kearnes et al., 2016; Ying et al., 2018; Wan et al., 2019). The GCN algorithms usually only consider the neighborhood with a short distance (e.g., the first-order neighborhood) because large distances will lead to over-smoothing, which degrades performance and increases computational complexity. However, the short distance easily leads to insufficient information about the neighborhood of the node (Li et al., 2018; Xu et al., 2018).

In this article, we propose a graph autoencoder approach for DTI prediction using a heterogeneous network (GADTI), which combines a graph convolutional network, matrix factorization, and random walk. GADTI first constructs a heterogeneous network that integrates eight data sources related to drugs and targets. Then, it runs a graph autoencoder model on the network to discover new DTIs. The encoder of the graph autoencoder includes two components: a GCN and a random walk with restart (RWR). The GCN component aggregates the first-order neighborhood information of each node and uses it to subsequently update the feature vector of nodes. The RWR component propagates the influence of nodes over the heterogeneous network. Through this, we obtain the embedding vectors of nodes, which are sent to the decoder. We use the matrix factorization model DistMult (Yang et al., 2015) to reconstruct the original heterogeneous network from the embedding vectors of nodes. Through the combination of GCN and RWR, GADTI can provide nodes with more information through a larger neighborhood while avoiding the over-smoothing and computational complexity caused by multi-layer message passing. The experimental results demonstrate that our approach is effective and efficient to predict potential DTIs.

MATERIALS AND METHODS

Dataset

We adopted a dataset used in previous studies (Luo et al., 2017; Wan et al., 2019). It consists of eight networks, including

TABLE 1 | Sources of datasets and their statistical information.

(a) Statistical information of nodes		
Node Type	Count	
Drug	708	
Targets	1,512	
Disease	5,603	
Side effect	4,192	
(b) Statistical information and source of edges		
Edge	Count	Data Source
drug–target interaction	1,923	DrugBank v3.0 (Knox et al., 2011)
Drug–drug interaction	10,036	DrugBank v3.0 (Knox et al., 2011)
Protein–protein	7,363	HPRD Release 9 (Keshava Prasad et al., 2009)
Drug–disease	199,214	Comparative Toxicogenomics Database (Davis et al., 2013)
Drug side effect	80,164	SIDER Version 2 (Kuhn et al., 2010)
Protein–disease	1,596,745	Comparative Toxicogenomics Database (Davis et al., 2013)
Drug structure similarity	*	Based on Morgan fingerprints (Rogers and Hahn, 2010)
Protein sequence similarity	*	Based on Smith–Waterman scores (Smith and Waterman, 1981)
Total	1,895,445	

*This edge is not counted because all node pairs are connected.

four types of nodes (drugs, targets, diseases, and side effects) and eight types of edges (drug–drug interaction, DTI, drug–disease association, drug–side effects association, protein–protein interaction, protein–disease association, drug chemical structure similarity, and protein sequence similarity). These data come from public databases such as DrugBank, HPRD, and SIDER. The weights of edges in all networks are non-negative. Furthermore, only the drug chemical structure similarity and the protein sequence similarity are real-valued, and thus represent drug–drug chemical structure similarity scores and protein–protein sequence similarity scores. The others are binary values indicating whether there is an interaction or association between nodes. **Table 1** lists the sources and statistics of these data.

Spatial-Based Graph Convolutional Network

Most recent network embedding methods are based on the GCN, especially spatial-based GCN. These methods define convolution on graph as neighborhood information aggregation. They generate embeddings for nodes by aggregating the local neighborhood of the nodes instead of the entire network, which is regarded as a message passing mechanism.

A typical spatial-based GCN method includes two phases. In the initialization phase, it generates an initial vector based

on the features of each node. If all the nodes in the network have no features, a one-hot vector is assigned to each node and a neural network is used to generate the initial vector. In the second phase, the vectors of nodes are updated by a combination of aggregated neighborhood vectors and the previous vectors of the nodes. These updates can be done through neural networks or linear transformations. The embedding vector of a node is a function of its neighborhood (including the node itself). This process looks similar to the receptive field of the convolution kernel in image processing, so it is called GCN. After one aggregation, the embedding vector of the node contains the feature information of its first-order neighbors. If we repeat this aggregation process K times, the embedding vector of the node can capture the feature information of its K -order neighbors. In the spatial-based GCN, the information of a node is first passed to its first-order neighbors, and then propagated to higher-order neighbors through edges on the network. Therefore, these methods are also called message passing methods. The process of graph convolution operation is summarized as follows:

$$\begin{aligned} a_v^{(n)} &= \text{AGGREGATE}^{(n)} \left(\left\{ h_u^{(n-1)} : u \in \mathcal{N}(v) \right\} \right), h_v^{(n)} \\ &= \text{UPDATE}^{(n)} \left(\left\{ h_v^{(n-1)}, a_v^{(n)} \right\} \right) \end{aligned} \quad (1)$$

where $\text{AGGREGATE}()$ and $\text{UPDATE}()$ are functions to aggregate neighborhood information and update node vectors, respectively; u, v are nodes; $a_v^{(n)}$ is the aggregated feature information of v at the n -th iteration; $\mathcal{N}(v)$ indicates the neighborhood of v ; and $h_v^{(n)}$ is the embedding vector of v at the n -th iteration. After the iteration, we obtain $h_v^{(K)}$, which represents the features of v and can be directly used for node-level tasks such as node similarity calculations, node classification, and link prediction.

Graph Autoencoder

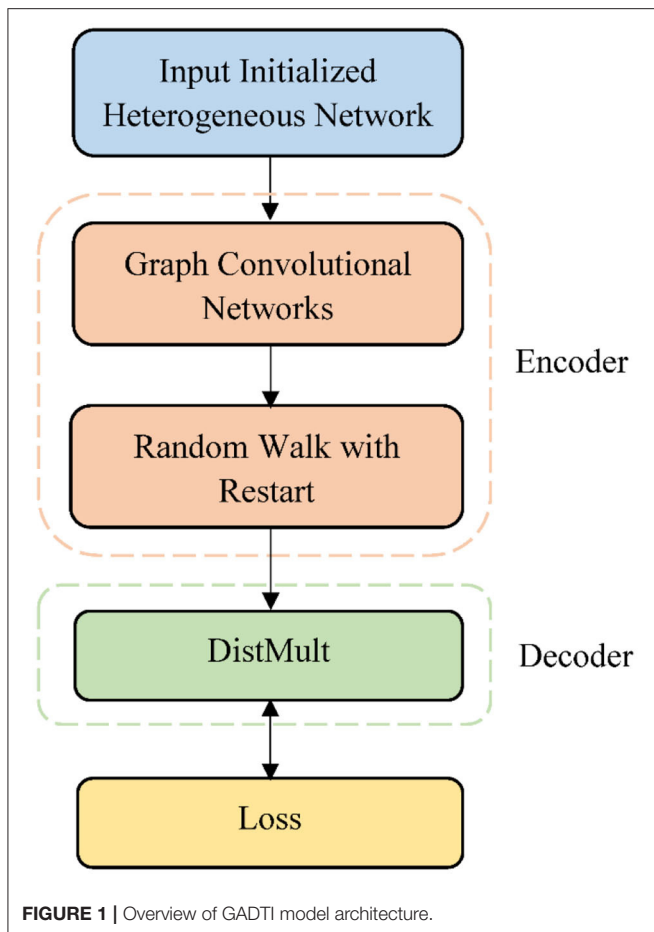
The graph autoencoder takes the network and the feature vectors of the node as input to generate a low-dimensional embedding vector of the node or the entire network.

Unlike traditional autoencoders, the encoder of a graph autoencoder is usually a GCN and its variants, and the decoder can be an inner product (Kipf and Welling, 2016; Pan et al., 2018) or matrix factorization (Zitnik et al., 2018; Lan et al., 2020). Generative adversarial networks (GANs) (Goodfellow et al., 2014) and attention mechanisms have also been applied to graph autoencoders (Ma et al., 2018; Pan et al., 2018; Jin et al., 2019). For heterogeneous graphs containing multiple edge types, the encoder aggregates neighbor features one by one according to the edge type, and then merges them to obtain the embedding vectors of the nodes (Gligorijevic et al., 2018; Ma et al., 2018; Zitnik et al., 2018).

GADTI

The data related to drugs and targets are represented in the form of a network, and the DTI prediction is then transformed into a link prediction of the network.

Definition 1 Network $G = (V, R)$, where $v \in V$ and $r \in R$ are nodes and edges, respectively.



Given a network G , v_d and v_t are the drug node and target node, respectively. Our goal is to determine whether the unknown edge $r_{dt} = (v_d, v_t)$ exists, or how likely it is to exist. To this end, we developed an end-to-end framework GADTI based on the graph autoencoder to discover new DTIs. This approach combines a graph convolutional network, matrix factorization, and random walk. GADTI first integrates multiple data sources to build a heterogeneous network, and then conducts prediction through a graph autoencoder model. As shown in **Figure 1**, GADTI has two main components:

- An encoder: a GCN followed by an RWR, which produces embeddings for nodes in G ;
- A decoder: a matrix factorization model using these embeddings to predict DTIs.

Encoder

The encoder consists of a GCN and an RWR. The GCN is used to aggregate first-order neighbor information to update node representation. Then, an RWR on the entire heterogeneous network allows the influence of nodes to spread far away so that we can obtain the final embedding vector. This approach can provide more information to nodes

through a larger neighborhood while avoiding the over-smoothing and computational complexity caused by multi-layer convolutional networks.

Aggregation by GCN

In this stage, only the first-order neighborhood of the node is considered. For each node, we first group its first-order neighbors according to the type of edge. Then, for each neighbor group, a neighborhood aggregation operation is performed to aggregate information. Finally, the neighbor information of different groups is accumulated and concatenated with the previous embedding vector of the node, and then sent to the neural network to generate a new embedding vector. The process of aggregating and updating are defined as follows:

$$\begin{aligned} a_v^r &= \sum_{u \in \mathcal{N}_r(v)} \frac{1}{c_r^v} \sigma(W_r^0 h_u^0 + b_r), h_v^* \\ &= \text{MEAN}(\{h_v^0\} \cup \{a_v^r : r \in R\}) \end{aligned} \quad (2)$$

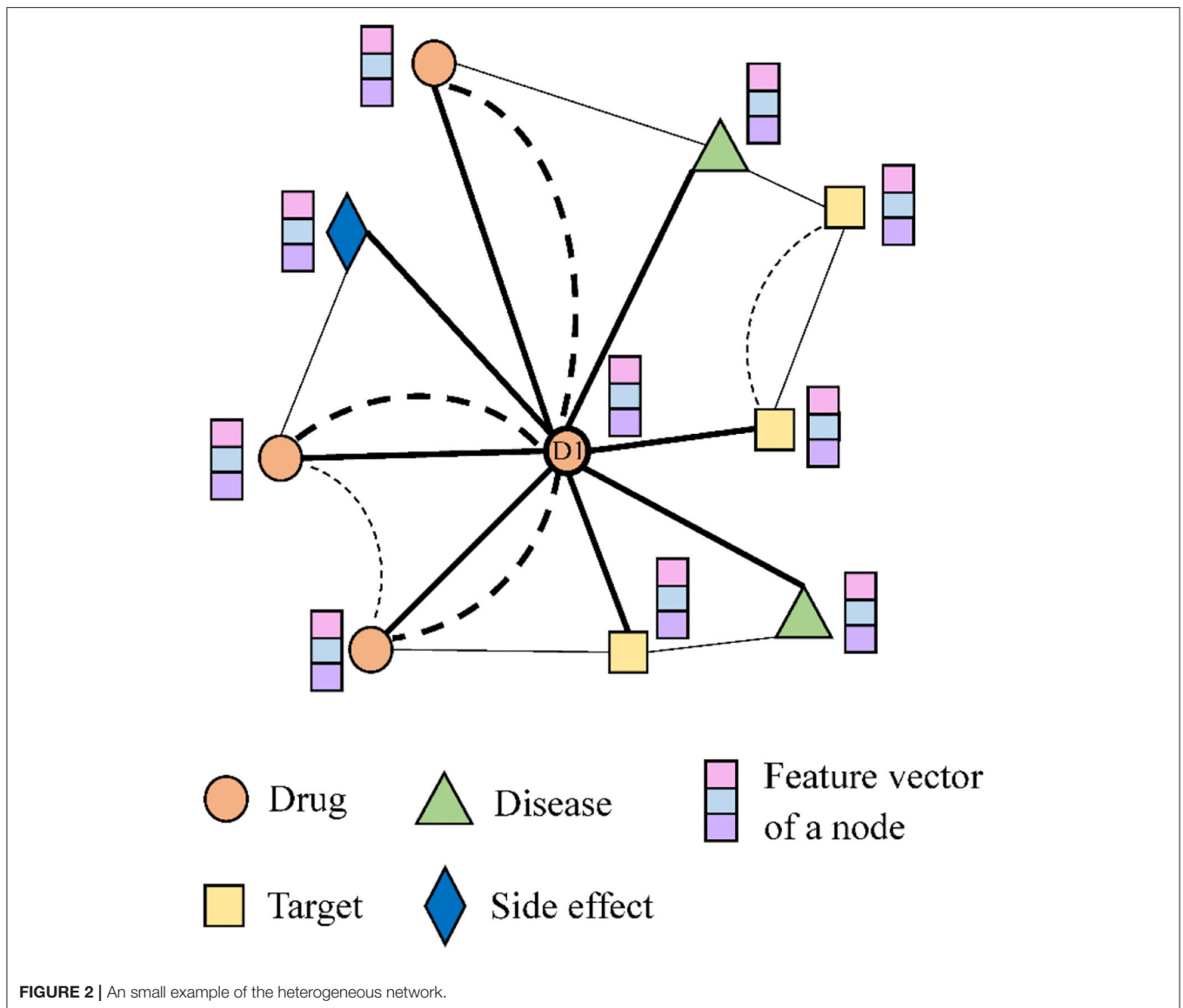
where a_v^r refers to the aggregated neighborhood information of v related to edge type r , $h_v^0 \in \mathbb{R}^d$ refers to the initial embedding vector of v , d denotes the dimension of vector, R indicates the set of edge types, $\mathcal{N}_r(v)$ are the neighbors of v related to edge type r , σ is a non-linear activation function, and $W_r^0 \in \mathbb{R}^{d \times d}$ and $b_r \in \mathbb{R}^d$ are edge-type specific parameter matrix and bias terms used to aggregate neighborhood information, respectively. c_r^v is a normalization constant that we choose to be $c_r^v = |\mathcal{N}_r(v)|$. $\text{MEAN}()$ is an element-wise mean operator, h_v^* is the updated embedding vector of v .

Figure 2 shows a small example of the network. Drug node $D1$ is associated with two diseases and one side effect, as well as targets two proteins, and interacts with three other drugs. The bold dotted line indicates the similarity between drugs.

The process of the encoder is provided in **Figure 3**. Multiple different single-layer neural networks (SLNs) are used in the encoder according to edge types. We take the drug node $D1$ in **Figure 2** as an example. Since there are five types of edges connected to $D1$, there will be five SLNs to aggregate neighbor information of corresponding edge types. The mean operator is chosen as the aggregation function, to perform an element-wise mean of the vectors in $\{h_v^0\} \cup \{a_v^r : r \in R\}$. It results in the new node embedding vector h^* . $\text{Relu}(x) = \max(0, x)$ is selected as the element-wise activation function. A projection with learnable parameters is employed to initialize h^0 .

Propagation by RWR

The multi-hop neighborhood information aggregation implemented by multi-layer convolution often leads to over-smoothing. The aforementioned GCN only considers the one-hop graph structure, which causes the multi-hop information of the node to be underutilized. In order to solve this problem, we introduce an RWR, which spreads the influence of nodes to other nodes that are not directly adjacent through a walk on the heterogeneous network. The introduction of multi-hop information extends the range of information aggregation from the first-order neighborhood to the high-order neighborhood, which is equivalent to increasing



the receptive field of convolution, thereby realizing long-range message passing.

Assuming that the transition matrix of the heterogeneous network is A and the restart probability is α , the RWR is defined as follows (Tong et al., 2008):

$$A_{ppr} = \alpha(I - (1 - \alpha)A)^{-1} \quad (3)$$

where I is the identity matrix, and $A_{ppr}(u, v)$ indicates the influence of node u on node v .

According to Equation (3), we can spread node information over long distances to get the final node embedding vector:

$$H^1 = \alpha(I - (1 - \alpha)A)^{-1}H^* \quad (4)$$

where H^* is the node embedding vector matrix obtained by the aforementioned convolution operation.

Since the time complexity of Equation (4) is $\mathcal{O}(n^2)$, when the network scale is large, it may be expensive. Therefore, we introduce the iterative form of Equation (4):

$$Z^0 = H^*, Z^{k+1} = (1 - \alpha)AZ^k + \alpha Z^0 \quad (5)$$

It is easy to prove that $\lim_{K \rightarrow \infty} Z^K = \alpha(I - (1 - \alpha)A)^{-1}H^*$.

Because all drug node pairs have edges of chemical structure similarity, there may be two edges between some drug node pairs. The same is true for target node pairs, and will bring inconvenience to the random walk. To simplify the problem, we delete the edges representing the similarity of drug structure and protein sequence from the heterogeneous network. That is, the graph convolution operates on a complete heterogeneous network whereas the random walk is only performed on a sub-network of the complete network.

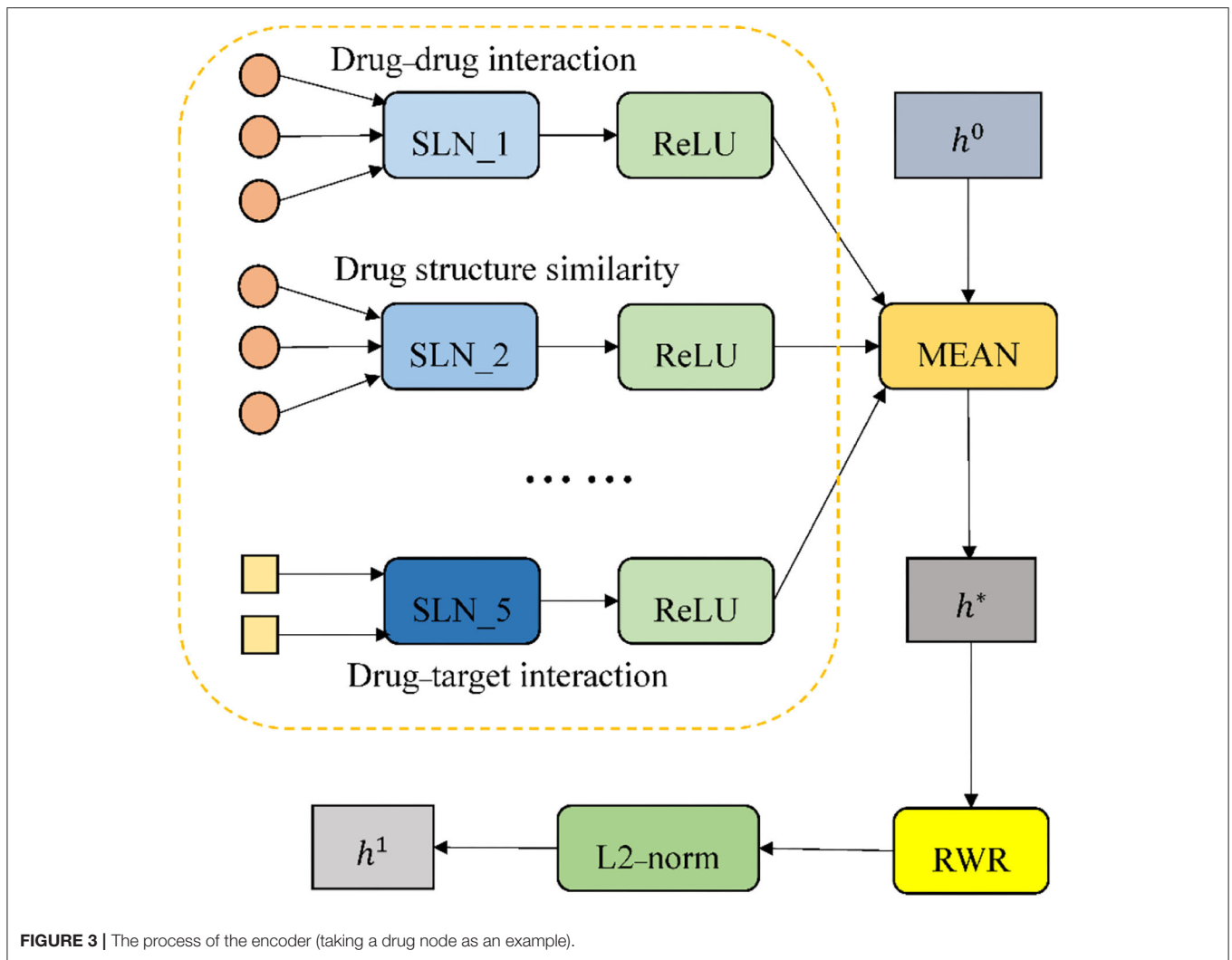


FIGURE 3 | The process of the encoder (taking a drug node as an example).

Decoder

While encoder maps each node in the heterogeneous network to a real-valued embedding vector, the decoder reconstructs the original network from the embedding vectors. The decoder is essentially a scoring function $s(u, r, v) : \mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$, used to score the triplets (u, r, v) so that we can evaluate the probability of edge r existing between u and v , where u and v are nodes, and r is a certain type of edge.

In our experiments, we use DistMult (Yang et al., 2015) as the decoder, which is known to perform well on standard link prediction benchmarks. The scoring function is:

$$s(u, r, v) = e_u^T M_r e_v \tag{6}$$

where e_u and e_v are the embedding vectors of u and v , respectively. e_u^T is the transpose of e_u , and $M_r \in \mathbb{R}^{d \times d}$ is an edge-type specific diagonal matrix.

In terms of Equation (6), we can reconstruct the original networks. Take the reconstruction of a DTI network as

an example:

$$DTI_{re} = V_{drug}^T M_{DTI} V_{protein} \tag{7}$$

where V_{drug} and $V_{protein}$ are the matrices of drug embedding vectors and target embedding vectors, respectively, and M_{DTI} is the diagonal matrix used to reconstruct the DTI network.

Training

The loss of network reconstruction is as follows:

$$\begin{aligned} \mathcal{L}_{re} &= \sum_{r \in \mathcal{R}} \sum_{i,j} (Q_{ij})^2, Q \\ &= P(\text{Network}_{original}^r - \text{Network}_{reconstruction}^r) \end{aligned} \tag{8}$$

where $\text{Network}_{original}^r$ and $\text{Network}_{reconstruction}^r$ are the original network with edge type r and the corresponding reconstructed network, respectively. P is a mask matrix where $P_{ij} = 1$ indicates that the element in the i -th row and j -th column of $\text{Network}_{original}^r$ appears in the training set, otherwise it does not occur. Q is a matrix that stores the difference between the predicted

value and the ground truth in the training set. We further add the regularization term of the weight coefficient to obtain the objective function:

$$\begin{aligned} \mathcal{L}_{re} &= \sum_{r \in R} \sum_{i,j} (Q_{ij})^2 + \lambda \sum_w w^2, Q \\ &= P(\text{Network}_{original}^r - \text{Network}_{reconstruction}^r) \end{aligned} \quad (9)$$

Our optimization goal is to minimize Equation (9), where $\sum_w w^2$ is the sum of the squares of all the weights, and λ is an adjustment coefficient. In GADTI, there are three trainable parameters: (1) four matrices for initializing node vectors, i.e., W^{drug} , $W^{disease}$, $W^{protein}$ and $W^{sideeffect}$; (2) 12 edge-type-specific neural network weight matrices W_r^0 for aggregating neighborhood information; and (3) 8 edge-type-specific diagonal matrices M_r used to reconstruct the networks.

We adopted the same sampling strategy and dataset division strategy as Wan et al. (2019). For the DTI network, the sample pair with an edge connection is regarded as the positive sample, and the sample pair without a connection is the negative sample. We randomly collect 10 negative samples for each positive sample to form the DTI dataset used by the model. Ten-fold cross-validation (Le et al., 2019) was used for performance evaluation. In each fold, the DTI dataset is randomly divided into three independent parts: training set, validation set and test set, with ratios of 0.855, 0.045, and 0.1 respectively. The training set of GADTI is composed of the training set of DTIs and other seven datasets. In each iteration, we update the model parameters on the training set, and then evaluate the model on the validation set. If the new model parameters show better performance on the validation set than before, the test set will be used to test the generalization ability of the model.

In addition to L2 regularization, early stopping is introduced to alleviate over-fitting. If the performance of the model on the validation set does not increase for n iterations, it can be considered that overfitting has occurred, so the training will stop early. Adaptive moment estimation algorithm (Adam) (Kingma and Ba, 2015) is selected to minimize the objective function. The dimension of embedding vector and the learning rate are set to 1,000 and 0.001, respectively, according to independent experiments. Our code runs on PyTorch V1.7 and DGL V0.5.

RESULTS

Performance Evaluation

We used 10-fold cross-validation to test the performance of our algorithm, and stratified sampling to ensure that the proportion of samples in each category in the training set and test set were the same as in the original dataset. The area under the receiver operator characteristic curve (AUROC) (Le, 2019) and the area under the precision–recall curve (AUPRC) were chosen to evaluate the performance of our approach and baseline methods.

The receiver operator characteristics (ROC) curve is suitable for evaluating the overall performance of the classifier because it takes both positive and negative samples into consideration (Le et al., 2020). However, class imbalance often occurs in actual datasets. For example, in a DTI network, the number of negative

samples is much larger than that of positive samples. In this case, the ROC curve presents an overly optimistic estimate of the effect. Conversely, both indicators of the precision–recall (PR) curve focus on positive samples. In the class imbalance cases, people are mainly concerned with positive samples, and thus the PR curve is widely considered to be better than the ROC curve. We use both AUROC and AUPRC. The larger the value of AUROC and AUPRC, the better the performance of the method.

Comparison With Baseline Methods

To evaluate the performance of GADTI, we compared it with four popular computational methods: MSCMF (Zheng et al., 2013), TL_HGBI (Wang et al., 2014), DTINet (Luo et al., 2017), and NeoDTI (Wan et al., 2019). These methods all predict DTIs from a heterogeneous network composed of multiple datasets. MSCMF uses matrix factorization methods and linear combinations of matrices to achieve prediction. TL_HGBI first establishes a three-layer heterogeneous network consisting of disease, drug, and protein data, and then uses an iterative strategy for drug repositioning. Meanwhile, DTINet focuses on learning low-dimensional vector representations of features that can accurately interpret the topological characteristics of each node in a heterogeneous network, and then makes predictions based on these representations through a vector space projection scheme. NeoDTI is close to the non-random walk version of GADTI. It first aggregates neighborhood information, and then reconstructs the network through two bilinear transformations. We run all five methods on the same dataset and implement three rounds of 10-fold cross-validation to compare their performance. The hyperparameters used in the baseline methods are the same as those in NeoDTI.

When the ratio of positive sample to negative sample is 1:10, the results of GADTI and the baseline methods are shown in **Figures 4, 5**. We observe that GADTI has an AUROC value of 0.9582, which is higher than those of NeoDTI (0.9509), DTINet (0.9208), TL_HGBI (0.8914), and MSCMF (0.8355). Meanwhile, in terms of AUPRC, which is more suitable for the current class imbalance case, GADTI is also better than all the baseline methods. Our approach slightly outperforms the second-best method (0.73% in terms of AUROC and 0.79% in terms of AUPRC).

Some DTI prediction methods based on machine learning include all unknown DTIs (treated as negative examples) in the training. To have a better comparison, we did additional test in this scenario. Experiment shows that GADTI still achieve the best performance, with an AUROC of 0.9369 and an AUPRC of 0.6205, and it stays ahead by a bigger margin. We notice that the AUROC values of all methods range from 0.8504 to 0.9369, but the AUPRC values range from 0.0312 to 0.6205, which is a large gap. **Figure 6** shows the experimental results of the dataset including all unknown DTIs.

The dataset in section Dataset contains homologous proteins or structurally similar drugs, which reduces the difficulty of predicting their interactions. In other words, the good performance of the DTI prediction method may come from a simple algorithm rather than a well-designed algorithm. Therefore, we carried out an additional experiment which is

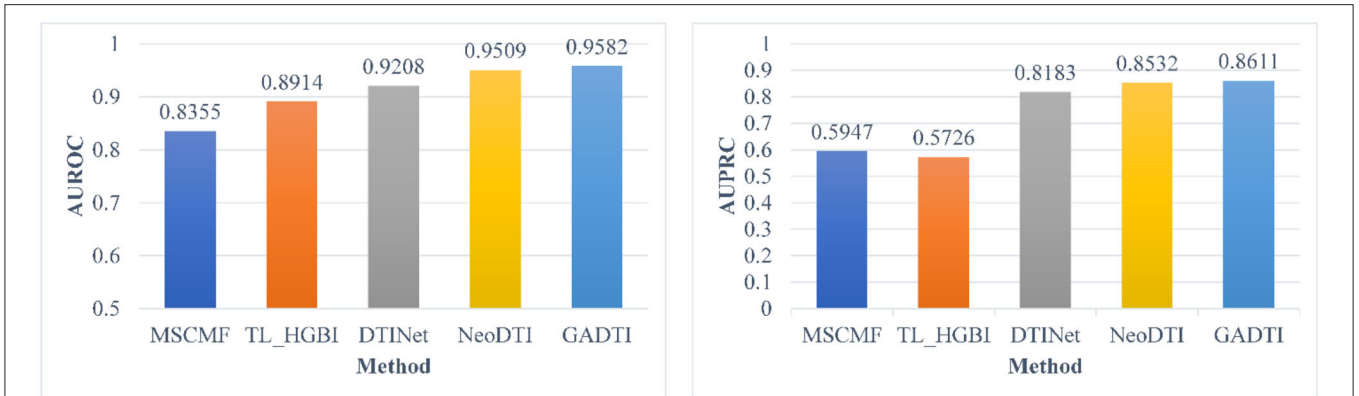


FIGURE 4 | Comparison between MSCMF, TL_HGBI, DTINet, NeoDTI, and GADTI in terms of AUROC and AUPRC based on 10-fold cross-validation (#positive: #negative = 1:10).

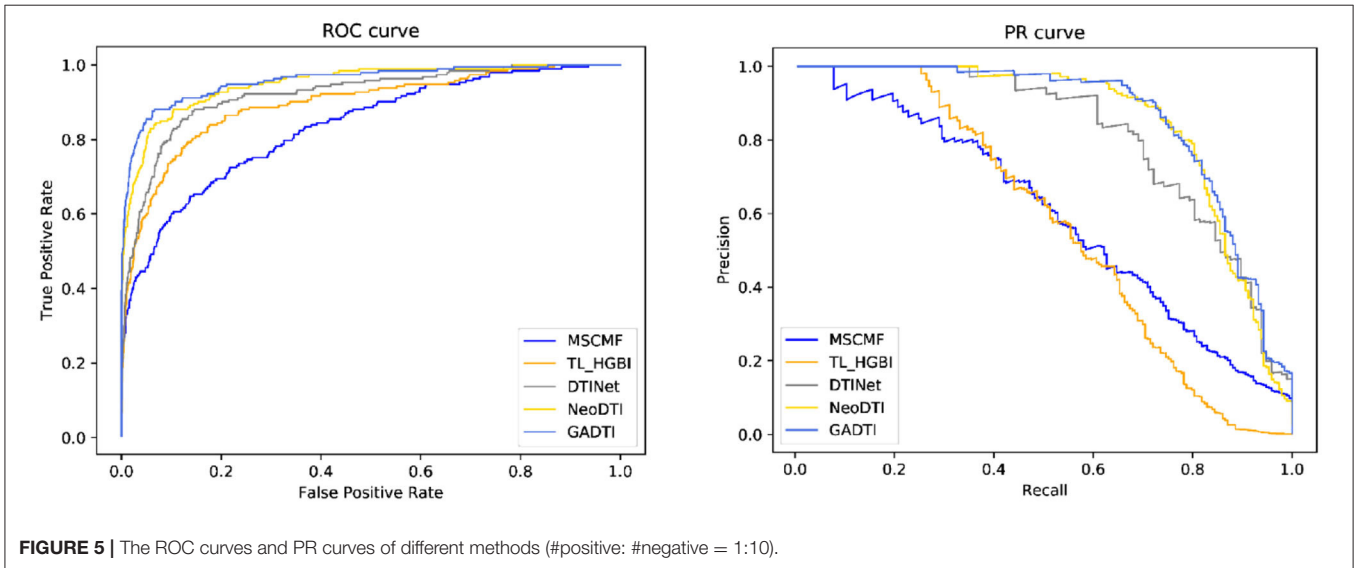


FIGURE 5 | The ROC curves and PR curves of different methods (#positive: #negative = 1:10).

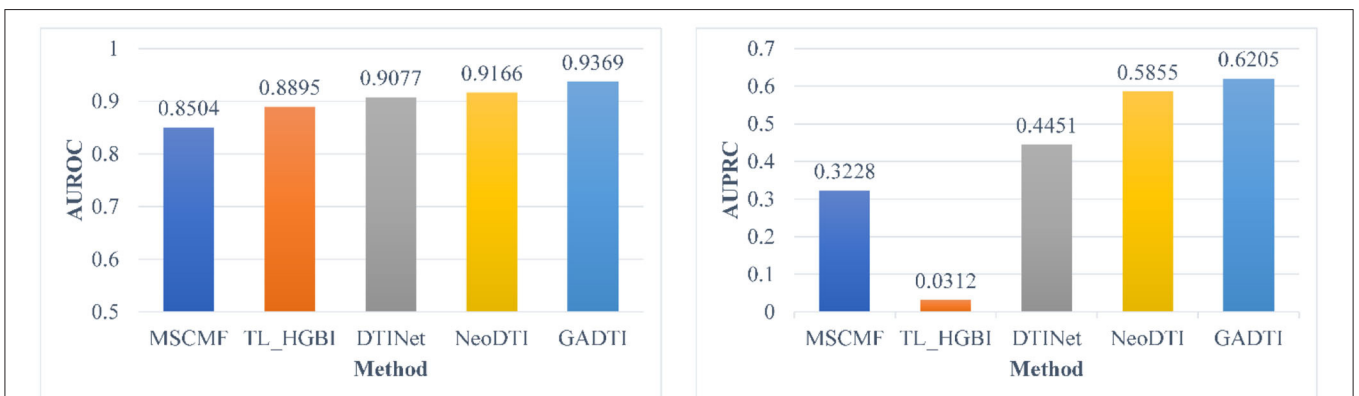


FIGURE 6 | Comparison between different methods in terms of AUROC and AUPRC based on 10-fold cross-validation (all unknown pairs were treated as negative examples).

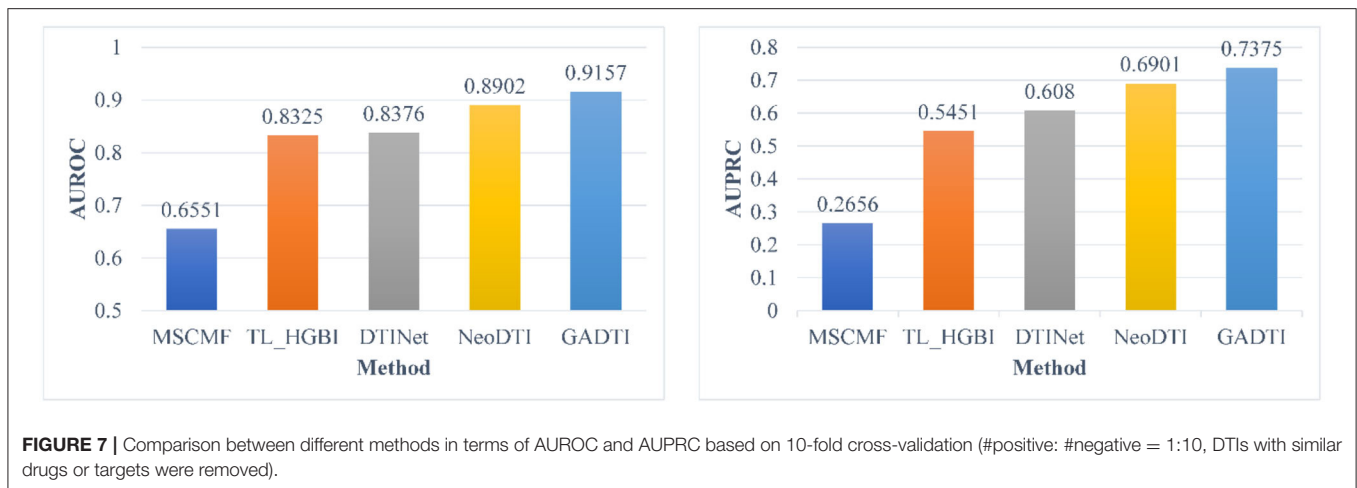


TABLE 2 | Hit numbers of GADTI in different configurations.

	<i>m</i> = 10	<i>m</i> = 20	<i>m</i> = 30	<i>m</i> = 40
Configuration A: #positive: #negative = 1:10	211	406	508	570
Configuration B: all unknown pairs were treated as negative examples	291	402	475	523
Configuration C: #positive: #negative = 1:10, and DTIs with similar drugs or targets were removed	149	265	351	422

the same as in Wan et al. (2019): the DTIs with homologous proteins (similarities > 0.4) or similar drugs (similarities > 0.6) were removed. **Figure 7** shows the experimental results, where the ratio of positive samples to negative samples is 1:10. GADTI greatly outperforms the second-best method (2.55% in terms of AUROC and 4.74% in terms of AUPRC).

Case Study

To evaluate the prediction performance, we downloaded the latest approved DTI dataset (V5.1.8, 2021-01-03) from DrugBank to verify the DTIs predicted by GADTI. We generated a set *DTI_newly*, which contained 1,040 new DTIs related to our original dataset from the latest DTI dataset. For each fold, top 40 potential DTIs were selected for each drug based on their predicted scores. Because the experiment used three rounds of 10-fold cross-validation, we obtained 30 tables with 708 rows and 40 columns. Each row represented the potential DTIs of a drug. The DTIs of each drug were then sorted in descending order by the number of occurrences. A predicted DTIs set *DTI_pre* was generated by selecting the top *m* DTIs for each drug. Finally, the number of DTIs (hit number) in the intersection of *DTI_newly* and *DTI_pre* was calculated to verify the reliability of the prediction. The results are shown in **Table 2**.

We observe that 54.8% of all new DTIs are predicted by GADTI in case of *m* = 40.

DISCUSSION

Finding novel DTI pairs is of great significance for drug development. However, biochemical experiments are very costly and time-consuming. Therefore, computational methods have attracted much attention recently because they can quickly and cheaply evaluate potential DTIs. Early DTI prediction studies are mainly divided into two categories: (a) inferring based on drug similarity and target similarity (Chen et al., 2012; Cheng et al., 2012; Mei et al., 2013; Wang et al., 2014; Seal et al., 2015); and (b) binary prediction based on drug feature and target feature (Nagamine et al., 2009; Lan et al., 2016; Olayan et al., 2018; Chen et al., 2019; Shi et al., 2019). The GADTI approach proposed in this paper also utilizes similarity data and the features of drugs and targets, which are represented in vectors. However, unlike previous studies, the network embedding method and the graph autoencoder framework are introduced to learn the embedding feature vectors of drugs and targets from multi-source heterogeneous networks for predicting unknown DTIs.

We use AUROC and AUPRC to evaluate the performance of GADTI and the baseline methods. The results show that GADTI greatly outperforms the other methods in three different scenarios. Only NeoDTI achieves comparable results under the situation where the ratio of positive sample to negative sample is 1:10 (**Figure 4**). This may be because NeoDTI also adopts GCN for aggregating and updating. In case study, GADTI accurately predicted 54.8% of the new DTIs (**Table 2**). We observe that the hit numbers of configuration B are less than those of configuration A, in case of *m* = 20, 30, and 40. However, the gap decreases with the decrease of *m*. We can see that in case of *m* = 10, the result is just reversed: the hit number of configuration B is much greater than that of configuration A. A reasonable inference is that configuration B, all unknown pairs are treated as negative examples, can make the ranking of potential DTIs more accurate. As a result of our experiments we conclude that, compared with baseline methods, GADTI is more reliable and effective in discovering potential DTIs. Hence, it can be used to identify new targets for existing drugs.

The reason why GADTI performs well is that it aggregates multi-hop neighborhood information and avoids over-smoothing. First of all, GADTI uses a GCN to aggregate first-order neighbor information from heterogeneous networks to update node representation. Then, an RWR is carried out on the whole network to spread the influence of nodes. The combination of the GCN and RWR introduces multi-hop information for node feature updating. It extends the scope of information aggregation from the first-order neighborhood to the higher-order neighborhood, which is equivalent to increasing the receptive field of convolution, thereby realizing long-range message passing.

Although GADTI has made outstanding achievements in DTI prediction, it still has room for improvement. For new nodes of drugs or targets that did not appear during training, GADTI cannot directly predict their interaction with known nodes, that is, it needs to restart training to make predictions. In addition, GADTI cannot predict isolated new nodes that are not associated with known drugs or target nodes. In future research, we will introduce node features and improve the model structure to try to solve these two problems.

REFERENCES

- Bagherian, M., Sabeti, E., Wang, K., Sartor, M. A., Nikolovska-Coleska, Z., and Najarian, K. (2020). Machine learning approaches and databases for prediction of drug-target interaction: a survey paper. *Brief. Bioinform.* 22, 247–269. doi: 10.1093/bib/bbz157
- Cai, H., Zheng, V. W., and Chang, C. C. (2017). A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Trans. Knowl. Data Eng.* 30, 1616–1637. doi: 10.1109/TKDE.2018.2807452
- Chen, Q., Lai, D., Lan, W., Wu, X., Chen, B., Chen, Y. P., et al. (2019). ILDMSF: inferring associations between long non-coding RNA and disease based on multi-similarity fusion. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/TCBB.2019.2936476. [Epub ahead of print].
- Chen, X., Liu, M. X., and Yan, G. Y. (2012). Drug-target interaction prediction by random walk on the heterogeneous network. *Mol. Biosyst.* 8, 1970–1978. doi: 10.1039/c2mb00002d
- Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., et al. (2012). Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput. Biol.* 8:e1002503. doi: 10.1371/journal.pcbi.1002503
- Davis, A. P., Murphy, C. G., Johnson, R., Lay, J. M., Lennon-Hopkins, K., Saraceni-Richards, C., et al. (2013). The comparative toxicogenomics database: update 2013. *Nucleic Acids Res.* 41, D1104–D1114. doi: 10.1093/nar/gks994
- Gligorijevic, V., Barot, M., and Bonneau, R. (2018). deepNF: Deep network fusion for protein function prediction. *Bioinformatics* 33, 3873–3881. doi: 10.1093/bioinformatics/bty440
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2* (Montreal, QC: MIT Press).
- Grover, A., and Leskovec, J. (2016). “node2vec: Scalable Feature Learning for Networks,” in *Conference on Knowledge Discovery and Data Mining*, eds. B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi (San Francisco, CA: ACM), 855–864.
- Jin, W., Yang, K. K., Barzilay, R., and Jaakkola, T. S. (2019). “Learning multimodal graph-to-graph translation for molecular optimization,” in *The 7th International Conference on Learning Representations: OpenReview.net* (New Orleans, LA).
- Karimi, M., Wu, D., Wang, Z., and Shen, Y. (2019). DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found at: <https://github.com/shulijiuba/GADTI>.

AUTHOR CONTRIBUTIONS

ZL and QC conceived the project, developed the prediction approach. ZL and WL designed and implemented the experiments. ZL, HP, XH, and SP analyzed the result. ZL wrote the paper. All authors read and approved the final manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (Nos. 61963004, 61702122 and 62072124), the Natural Science Foundation of Guangxi (Nos. 2017GXNSFDA198033 and 2018GXNSFBA281193), the Key Research and Development Plan of Guangxi (No. AB17195055), and the Science and Technology Base and talent Special project of Guangxi (No. AD20159044).

- and convolutional neural networks. *Bioinformatics* 35, 3329–3338. doi: 10.1093/bioinformatics/btz111
- Kearnes, S., Mccloskey, K., Berndl, M., Pande, V., and Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.* 30, 1–14. doi: 10.1007/s10822-016-9938-8
- Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., et al. (2009). Human protein reference database 2009 update. *Nucleic Acids Res.* 37, D767. doi: 10.1093/nar/gkn892
- Kingma, D. P., and Ba, J. (2015). Adam: a method for stochastic optimization. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1412.6980v9> (accessed May 15, 2020).
- Kipf, T. N., and Welling, M. (2016). Variational graph auto-encoders. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1611.07308> (accessed September 9, 2019).
- Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., et al. (2011). DrugBank 3.0: a comprehensive resource for ‘omics’ research on drugs. *Nucleic Acids Res.* 39, D1035–1041. doi: 10.1093/nar/gkq1126
- Kuhn, M., Campillos, M., Letunic, I., Jensen, L. J., and Bork, P. (2010). A side effect resource to capture phenotypic effects of drugs. *Mol. Syst. Biol.* 6:343. doi: 10.1038/msb.2009.98
- Lan, W., Lai, D., Chen, Q., Wu, X., Chen, B., Liu, J., et al. (2020). LDICDL: LncRNA-disease association identification based on Collaborative Deep Learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/TCBB.2020.3034910. [Epub ahead of print].
- Lan, W., Wang, J., Li, M., Liu, J., Li, Y., Wu, F.-X., et al. (2016). Predicting drug-target interaction using positive-unlabeled learning. *Neurocomputing* 206, 50–57. doi: 10.1016/j.neucom.2016.03.080
- Le, N. Q. K. (2019). Fertility-GRU: identifying fertility-related proteins by incorporating deep-gated recurrent units and original position-specific scoring matrix profiles. *J. Proteome Res.* 18, 3503–3511. doi: 10.1021/acs.jproteome.9b00411
- Le, N. Q. K., Do, D. T., Chiu, F. Y., Yapp, E. K. Y., Yeh, H. Y., and Chen, C. Y. (2020). XGBoost improves classification of MGMT promoter methylation status in IDH1 wildtype glioblastoma. *J. Pers. Med.* 10:128. doi: 10.3390/jpm10030128
- Le, N. Q. K., Yapp, E. K. Y., Nagasundaram, N., and Yeh, H. Y. (2019). Classifying promoters by interpreting the hidden information of DNA sequences via deep learning and combination of continuous fasttext N-grams. *Front. Bioeng. Biotechnol.* 7:305. doi: 10.3389/fbioe.2019.00305

- Li, Q., Han, Z., and Wu, X. M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1801.07606v1> (accessed April 20, 2020).
- Liu, Z., Chen, Q., Lan, W., Liang, J., Chen, Y.-P. P., and Chen, B. (2020). A survey of network embedding for drug analysis and prediction. *Curr. Protein Peptide Sci.* 21:1. doi: 10.2174/1389203721666200702145701
- Luo, Y., Zhao, X., Zhou, J., Yang, J., Zhang, Y., Kuang, W., et al. (2017). A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun.* 8:573. doi: 10.1038/s41467-017-00680-8
- Ma, T., Xiao, C., Zhou, J., and Wang, F. (2018). “Drug similarity integration through attentive multi-view graph auto-encoders,” in *The 27th International Joint Conference on Artificial Intelligence*, ed. J. Lang (California: International Joint Conferences on Artificial Intelligence), 3477–3483.
- Mei, J., Kwok, C. K., Yang, P., Li, X., and Zheng, J. (2013). Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics* 29, 238–245. doi: 10.1093/bioinformatics/bts670
- Mohamed, S. K., Nováček, V., and Nounu, A. (2019). Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics* 36, 603–610. doi: 10.1093/bioinformatics/btz600
- Nagamine, N., Shirakawa, T., Minato, Y., Torii, K., Kobayashi, H., Imoto, M., et al. (2009). Integrating statistical predictions and experimental verifications for enhancing protein-chemical interaction predictions in virtual screening. *PLoS Comput. Biol.* 5:e1000397. doi: 10.1371/journal.pcbi.1000397
- Olayan, R. S., Ashoor, H., and Bajic, V. B. (2018). DDR: Efficient computational method to predict drug-target interactions using graph mining and machine learning approaches. *Bioinformatics* 34, 1164–1173. doi: 10.1093/bioinformatics/btx731
- Öztürk, H., Ozkirimli, E., and Özgür, A. (2019). WideDTA: prediction of drug-target binding affinity. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1902.04166v1> (accessed April 8, 2020).
- Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., and Zhang, C. (2018). “Adversarially regularized graph autoencoder for graph embedding,” in: *The 27th International Joint Conference on Artificial Intelligence*, ed. J. Lang (California: International Joint Conferences on Artificial Intelligence), 2609–2615.
- Perozzi, B., Alrfou, R., and Skiena, S. (2014). “DeepWalk: online learning of social representations,” in: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds. S.A. Macskassy, C. Perlich, J. Leskovec, W. Wang and R. Ghani (New York, NY: ACM), 701–710.
- Rogers, D., and Hahn, M. (2010). Extended-connectivity fingerprints. *J Chem Inf Model* 50, 742–754. doi: 10.1021/ci100050t
- Seal, A., Ahn, Y.-Y., and Wild, D. J. (2015). Optimizing drug-target interaction prediction based on random walk on heterogeneous networks. *J. Cheminform.* 7:40. doi: 10.1186/s13321-015-0089-z
- Shi, H., Liu, S., Chen, J., Li, X., Ma, Q., and Yu, B. (2019). Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure. *Genomics* 111, 1839–1852. doi: 10.1016/j.ygeno.2018.12.007
- Smith, T. F., and Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197. doi: 10.1016/0022-2836(81)90087-5
- Su, C., Tong, J., Zhu, Y., Cui, P., and Wang, F. (2018). Network embedding in biomedical data science. *Brief. Bioinform.* 21, 182–197. doi: 10.1093/bib/bby117
- Tong, H., Faloutsos, C., and Pan, J. Y. (2008). Random walk with restart: fast solutions and applications. *Knowl. Inform. Syst.* 14, 327–346. doi: 10.1007/s10115-007-0094-2
- Wan, F., Hong, L., Xiao, A., Jiang, T., and Zeng, J. (2019). NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug-target interactions. *Bioinformatics* 35, 104–111. doi: 10.1093/bioinformatics/bty543
- Wang, W., Yang, S., Zhang, X., and Li, J. (2014). Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics* 30, 2923–2930. doi: 10.1093/bioinformatics/btu403
- Wen, M., Zhang, Z., Niu, S., Sha, H., Yang, R., Yun, Y., et al. (2017). Deep-learning-based drug-target interaction prediction. *J. Proteome Res.* 16, 1401–1409. doi: 10.1021/acs.jproteome.6b00618
- Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., and Jegelka, S. (2018). “Representation learning on graphs with jumping knowledge networks,” in: *The 35th International Conference on Machine Learning*, eds. J.G. Dy and A. Krause (UK: PMLR), 5449–5458.
- Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2015). “Embedding Entities and Relations for Learning and Inference in Knowledge Bases,” in: *The 3rd International Conference on Learning Representations*, eds. Y. Bengio and Y. LeCun (ICLR). Available online at: www.iclr.cc
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). “Graph convolutional neural networks for web-scale recommender systems,” in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (London: Association for Computing Machinery)*, 974–983.
- Zheng, X., Ding, H., Mamitsuka, H., and Zhu, S. (2013). “Collaborative matrix factorization with multiple similarities for predicting drug-target interactions,” in: *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds. I.S. Dhillon, Y. Koren, R. Ghani, T.E. Senator and P. Bradley (New York, NY: ACM), 1025–1033.
- Zhu, S., Bing, J., Lin, C., Zeng, X., and Min, X. (2018). Prediction of drug-gene interaction by Using Metapath2vec. *Front. Genet.* 9:248. doi: 10.3389/fgene.2018.00248
- Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 457–466. doi: 10.1093/bioinformatics/bty294
- Zong, N., Kim, H., Ngo, V., and Harismendy, O. (2017). Deep mining heterogeneous networks of biomedical linked data to predict novel drug-target associations. *Bioinformatics* 33, 2337–2344. doi: 10.1093/bioinformatics/btx160
- Zong, N., Wong, R. S. N., Ngo, V., Yu, Y., and Li, N. (2019). Scalable and accurate drug-target prediction based on heterogeneous bio-linked network mining. *bioRxiv [Preprint]*. doi: 10.1101/539643. Available online at: <https://www.biorxiv.org/content/10.1101/539643v1> (accessed April 10, 2020).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Liu, Chen, Lan, Pan, Hao and Pan. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.