AutoDDI: Drug–Drug Interaction Prediction With Automated Graph Neural Network

Jianliang Gao, Zhenpeng Wu, Raeed Al-Sabri, Babatounde Moctard Oloulade, and Jiamin Chen, *Student Member, IEEE*

***Abstract*—Drug–drug interaction (DDI) has attracted widespread attention because when incompatible drugs are taken together, DDI will lead to adverse effects on the body, such as drug poisoning or reduced drug efficacy. The adverse effects of DDI are closely determined by the molecular structures of the drugs involved. To represent drug data effectively, researchers usually treat the molecular structure of drugs as a molecule graph. Then, previous studies can use the handcrafted graph neural network (GNN) model to learn the molecular graph representations of drugs for DDI prediction. However, in the field of bioinformatics, manually designing GNN architectures for specific molecular structure datasets is time-consuming and depends on expert experience. To address this problem, we propose an automatic drug–drug interaction prediction method named AutoDDI that can efficiently and automatically design the GNN architecture for drug–drug interaction prediction without manual intervention. To this end, we first design an effective search space for drug–drug interaction prediction by revisiting various handcrafted GNN architectures. Then, to efficiently and automatically design the optimal GNN architecture for each drug dataset from the search space, a reinforcement learning search algorithm is adopted. The experiment results show that AutoDDI can achieve the best performance on two real-world datasets. Moreover, the visual interpretation results of the case study show that AutoDDI can effectively capture drug substructure for drug–drug interaction prediction.**

***Index Terms*—Drug–drug interaction, graph neural network, graph neural architecture search, reinforcement learning.**

I. Introduction

WHEN two or more incompatible drugs are taken together, drug–drug interaction (DDI) will lead to adverse effects on the body [1], [2], such as drug poisoning or reduced drug efficacy. This phenomenon of DDI has attracted widespread attention. However, many patients need to use multiple drugs simultaneously to treat complex diseases [2]. Therefore, while patients undergo multi-drug treatment, DDI increases the risk of adverse effects and treatment failure. DDI is crucial for the safety and effectiveness of drug treatment, which has prompted many efforts to identify whether DDI occurs when two given drugs are taken together.

Identifying DDI remains a challenging task because the massive number of drug combinations makes experimental tests and clinical trials very expensive and almost impossible [3]. To alleviate the challenge of identifying DDI, many works have begun to explore an effective and alternative approach, namely computational methods [4], [5], [6]. Computational methods usually assume that drug pairs with similar features tend to generate similar DDI. Based on the knowledge distilled from existing DDI, computational methods predict potential DDI. To fully and effectively mine the raw features of drugs, such as the molecular structure, recent work mainly focuses on using deep neural networks [7], [8], in which graph neural networks (GNNs) have shown promising performance in learning the graph representation of drug molecules [9], [10], [11], [12], [13]. GNN-based DDI prediction methods first represent the molecular structure of each drug into a graph, and learn the graph representation (or drug substructure) of each drug in the drug pair. Then, the graph representation of each drug and a co-attention mechanism [10], [11], [12], [13] are used to predict DDI. Among GNN-based DDI prediction methods, most methods only separately learn the graph representation from each drug, not considering the interaction information stored in the drug pair. Recent methods, including SSI-DDI [11], GMPNN-CS [12], SA-DDI [14] and so on, integrate interactions at graph-level in the drug pair, thus improving performance compared to previous methods. However, the interaction information is usually derived from the final global pooling of those methods, they overlook the detailed interactions between atoms in the drug pair, which provide more valuable information. DSN-DDI increases the completeness of drug information by aggregating interactions between atoms in the drug pair [13], thereby improving the accuracy of DDI prediction.

Although GNN-based approaches have achieved success in the DDI prediction task, they fix the operation of each architecture component to encode the graph representation of drug molecules, resulting in performance loss. To demonstrate this problem, we visualize the model performance of different operations under the convolution architecture component in Fig. 1. As shown in Fig. 1, the convolution operation with

Manuscript received 19 October 2023; revised 8 December 2023; accepted 26 December 2023. Date of publication 4 January 2024; date of current version 7 March 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3603000 and in part by the National Natural Science Foundation of China under Grant 62272487. *(Corresponding author: Jiamin Chen.)*

The authors are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: gaojianliang@csu.edu.cn; zhenpeng@csu.edu.cn; alsabriraeed@csu.edu.cn; olouladem octard@csu.edu.cn; chenjiamin@csu.edu.cn).

The code is available at: https://github.com/Zhen-Peng-Wu/AutoDDI.

Digital Object Identifier 10.1109/JBHI.2024.3349570

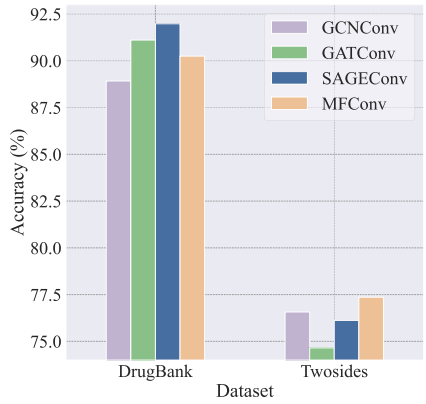

Fig. 1. Convolution operation with the best performance varies for different datasets. Thus, adapting the convolution operation for each dataset helps to improve the model performance in drug–drug interaction prediction. In this experiment, the number of GNN layers is fixed as two, and the performance indicator is classification accuracy.

the best performance varies for different datasets. Therefore, adapting the convolution operation for each dataset will help to improve the model performance in drug–drug interaction prediction. However, manually adapting the operation of all architecture components for each drug dataset is time-consuming and depends on expert experience. To achieve the automatic design of GNN architecture, a lot of work has focused on graph neural architecture search (GNAS) [15], [16], [17].

In this work, to automatically design the optimal GNN architecture for each drug dataset, we propose drug–drug interaction prediction with automated graph neural network (AutoDDI) with the help of GNAS. First, an effective drug–drug interaction prediction search space for encoding the drug graph representation is designed. Then, AutoDDI uses reinforcement learning search to automatically identify the optimal GNN architecture from the drug–drug interaction prediction search space with little manual intervention. To verify the effectiveness of AutoDDI, we evaluate AutoDDI on two real-world datasets, DrugBank and Twosides. We also evaluate AutoDDI under two settings: transductive and inductive. Compared with previous handcrafted GNN architectures, AutoDDI has achieved the best performance on different datasets and settings. Briefly, the main contributions of this work can be summarized as follows:

- We propose AutoDDI that can automatically identify the optimal GNN architecture to obtain the effective drug substructure representation for drug–drug interaction prediction. To the best of our knowledge, AutoDDI is the first attempt to design GNN architecture for drug–drug interaction prediction task automatically.
- We design a suitable and effective drug–drug interaction prediction search space, in which we add GNN layers architecture component to automatically adapt GNN depth and bipartite convolution architecture component to capture interaction information of a drug pair.
- We conduct extensive experiments using different evaluation metrics based on the two widely used datasets. The experiment results demonstrate that AutoDDI can achieve better performance than previous handcrafted GNN architectures.

II. Related Work

A. Drug–Drug Interaction Prediction

Computational methods for DDI prediction can be broadly divided into two categories: (1) all drugs form a network; (2) each drug independently forms a graph.

For the network-based approach, researchers assume that drugs form an interconnected system, where nodes represent drugs, and edges indicate the similarity between drug pair [18], [19] or indicate the DDI between drug pair [20], [21], [22], [23], [24]. The network-based approach uses different algorithms to predict potential DDI from derived networks, including label propagation [18], matrix factorization [4], [20], [22], and deep auto-encoders [19], [24]. The network-based approach improves prediction performance by adding additional topological information about the drug interconnected system to the model. This transductive learning method of observing all data beforehand limits the use of the network-based approach.

For the GNN-based approach, researchers assume that each drug forms a graph based on the molecular structure of the drug itself, where nodes represent atoms, and edges indicate the bond between two atoms [9], [10], [11], [12], [13]. The GNN-based approach can perform DDI prediction for unseen drug pairs, therefore it belongs to inductive learning. The GNN-based approach learns the graph representation of each drug, and weights the graph representation of each drug using a co-attention mechanism [10], [11], [12], [13]. Then, the weighted graph representations of each drug are aggregated to predict DDI between the drug pair. Among GNN-based DDI prediction methods, most methods only separately learn the graph representation from each drug, not considering the interaction information stored in the drug pair. Recent methods, including SSI-DDI [11], GMPNN-CS [12], SA-DDI [14] and so on, integrate interactions at graph-level in the drug pair, thus improving performance compared to previous methods. However, the interaction information is usually derived from the final global pooling of those methods, they overlook the detailed interactions between atoms in the drug pair, which provide more valuable information. DSN-DDI increases the completeness of drug information by aggregating interactions between atoms in the drug pair [13], thereby improving the accuracy of DDI prediction. Although GNN-based approaches have achieved success in the DDI prediction task, designing a GNN architecture for a special drug dataset depends on expert experience and is time-consuming. Therefore, this indicates a demand for automatically designing the GNN architecture for a special drug dataset.

B. Capturing Drug Substructure by Stacking GNN Layers

In the graph classification task such as DDI prediction, the GNN architecture needs to capture information from long-distant neighbors [25], which helps to improve the representation learning ability of model [17], [26], [27]. Deeper GNNs can capture information from long-distant neighbors by the larger receptive field. However, as the number of GNN layers increases,

it will suffer from the over-smoothing problem [28], [29] due to the node features indistinguishable. Recent work [17] has justified that the over-smoothing problem has a smaller influence on the graph classification task than the node classification task, and stacking GNN layers is a feasible solution for capturing information from long-distant neighbors. Stacking GNN layers is a direct and sufficient way to capture information from long-distant neighbors while maintaining the graph topology unchanged.

Therefore, for DDI prediction, the drug substructure captured by the GNN-based approach is equivalent to the captured information from long-distant neighbors [13], [17]. A drug is an entity made up of different chemical substructures [30], which jointly decide the pharmacological properties of the drug and interaction types. The drug substructure provides practical meaning for the captured information from long-distant neighbors. Capturing the drug substructures is the key factor in DDI prediction. The experimental results of [12] show that extracting drug substructures by other ways is deficient. The GNN-based approach with good performance extracts drug substructures by stacking GNN layers [11], [13]. However, these methods do not automatically adapt to the number of GNN layers for each dataset.

C. Graph Neural Architecture Search

A large number of works seek to automatically design GNN architectures using graph neural architecture search (GNAS) [15], [16], [17], [31]. The entire process of GNAS is usually made up of four steps. The first step is to design a suitable search space with different GNN architecture components based on the corresponding task. For instance, the search space on the node classification task includes convolution function component, hidden dimension component, and activation function component. Each architecture component holds many candidate operations, and the combination of different architecture components further generates GNN architectures, such as $GATConv$, 64, $Relu$. The second step is to sample GNN architectures within the search space by the search algorithm and then train sampled GNN architectures on the training set. The third step is to evaluate sampled GNN architectures using the estimation strategy based on the validation set and generate the estimation feedback. Finally, the estimation feedback is used to guide the search algorithm iteration.

The existing GNAS works mainly make an effort to the node classification task. Most of these works have achieved the automatic design of aggregation layers. For instance, GraphNAS [32], AutoGM [33], and DSS [34] consider designing aggregation layers with various components, such as convolution function, activate function and hidden dimension, etc. Besides, AutoGraph [35], SNAG [36], SANE [37], and F2GNN [38] automatically search the skip connections. Apart from designing aggregation layers, the works for the graph classification task also require designing pooling operations. For example, RE-MPNN [39] automatically designs global pooling operations additionally, and PAS [15] automatically learns local and global pooling operations. Although these works have been successful, they are not usually designed for bipartite graphs, while AutoDDI provides one search space for bipartite graphs constructed by drug pairs.

III. Method

In this section, the problem we are attempting to solve is first mathematically formulated. Then, we describe the complete process of automatically designing the optimal GNN. Finally, the optimization objective of drug–drug interaction prediction is introduced.

A. Problem Definition

The graph neural architecture search is a bi-level optimization problem and is mathematically formulated as follows:

$$\begin{aligned} m_{opt} &= \underset{(m\in M)}{\operatorname{argmax}}\, R(m(w^*), D_v) \\ s.t.\quad w^* &= \underset{w}{\operatorname{argmin}}\, \mathcal{L}(m(w), D_t) \end{aligned} \tag{1}$$

where M represents the search space in neural architecture search, R is the estimation strategy that generates an objective to guide the search algorithm iteration to obtain the model with better performance, D_t and D_v are the training and validation sets, respectively. Graph neural architecture search aims to identify the optimal model m_{opt} from search space M, which is trained on D_t based on loss function $\mathcal{L}$ and obtains the best performance on D_v based on the estimation strategy R.

B. Overview of AutoDDI

In order to better understand our proposed method AutoDDI, we will illustrate the workflow of AutoDDI by Fig. 2. The drug–drug interaction prediction based on AutoDDI contains two main stages. Stage 1 is automated graph neural network design, which is built on the drug–drug interaction prediction task. This stage follows the typical process of GNAS: Using the reinforcement learning search algorithm to sample a GNN architecture from the search space, where the search space is designed based on the drug–drug interaction prediction task; The sampled model will be trained based on the drug–drug interaction dataset and evaluated on the validation set based on the estimation strategy; The validation performance is used as a feedback signal to guide the iteration of the reinforcement learning search algorithm. When the reinforcement learning search is completed, each GNN architecture component will obtain the optimal candidate operation, and the optimal candidate operations of all GNN architecture components constitute the optimal GNN architecture. Stage 2 is drug–drug interaction prediction, which retrains the optimal GNN architecture of AutoDDI automatically based on the drug–drug interaction dataset. In this stage, the optimal GNN architecture will obtain an effective representation of a drug by aggregating the information from the drug graph itself and the bipartite graph construed by two drugs. Finally, AutoDDI predicts whether DDI exists between two drugs using the MLP with the drug representation of all GNN layers of two drugs.

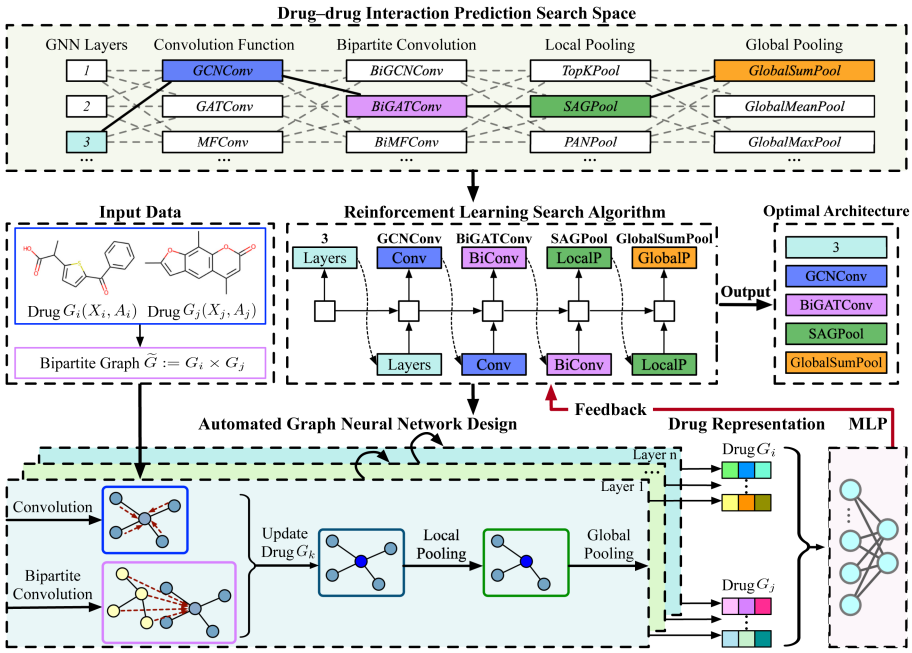

Fig. 2. AutoDDI framework. The drug–drug interaction prediction task based on AutoDDI includes two stages. Stage 1 is automated graph neural network design, which automatically learns the optimal GNN architecture from drug–drug interaction prediction search space by reinforcement learning search algorithm. Reinforcement learning generates a sequence description of all architecture components, where each sequence element represents a candidate operation corresponding to each architecture component. The child GNN architecture will be built based on the sequence description of architectures and trained on the training set. The validation performance of the child GNN architecture will be used as a feedback signal for reinforcement learning iterations. Stage 2 is drug–drug interaction prediction, which automatically builds the optimal GNN architecture to generate drug representation by encoding the drug molecular structure. Then, AutoDDI uses drug representations of all layers of GNN as MLP input for predicting DDI probability between a drug pair.

C. Automated Graph Neural Network Design

In this work, the process of automated GNN design is as follows: First, we construct the search space based on the drug–drug interaction prediction task; Second, the reinforcement learning search algorithm is used to identify the optimal GNN architecture from the DDI prediction search space, where the DDI classification accuracy serves as the estimation strategy guiding reinforcement learning iterations; Finally, the optimal GNN architecture will be automatically constructed after completing the reinforcement learning search. In this section, we will mainly introduce the design of the drug–drug interaction prediction search space and the detailed progress of the reinforcement learning search algorithm.

1) Drug–Drug Interaction Prediction Search Space: In an effort to identify a GNN architecture with great performance for the DDI prediction task, we design a drug–drug interaction prediction search space with advanced candidate operations of different GNN architecture components based on the DDI prediction task. The architecture components and their candidate operations are as follows:

GNN Layers: The over-smoothing problem has a smaller influence on the graph classification task [17] than the node classification task. Information from long-distant neighbors can be captured by increasing the number of GNN layers [25], which helps improve the predictive ability of GNN architectures [17], [27]. In the DDI prediction task, the GNN architecture needs to capture information from long-distant neighbors to identify drug substructure [13], which is a crucial factor for DDI prediction. However, blindly increasing the number of GNN layers cannot achieve the highest performance of the GNN architecture, and the GNN architecture may reach the highest performance at the middle GNN layer [17]. Hence, the GNN architecture requires an adaptive GNN layer scheme instead of a fixed GNN layer. Although blindly increasing the number of GNN layers is not advisable, sufficient GNN layers are required. At the same time, we need to consider the computational cost of the GNN architecture. Therefore, AutoDDI will automatically learn the number of GNN layers, and the candidate operations of GNN layers in this work include 1, 2, 3, 4, 5, 6, 7, and 8.

Convolution Function: For graph convolution, the representation of the central node relies on its neighbor nodes. The features of different neighbor nodes leads to diverse effects on the central node. Specifically, graph convolution calculates coefficients of a central node with its neighbor nodes [16] and then aggregates the features of neighbor nodes with calculated coefficients to form the representation of the central node [16]. In this work, we provide seven graph convolutions using the Pytorch-Geometric library [40]. The candidate operations of the convolution function are as follows: *GCNConv* [41], *GATConv* [42], *GraphConv* [43], *GeneralConv* [44], *MFConv* [45], *LEConv* [46] with sum aggregator, *SAGEConv* [47] with mean aggregator.

TABLE I
CANDIDATE OPERATIONS OF ARCHITECTURE COMPONENTS IN DRUG–DRUG INTERACTION PREDICTION SEARCH SPACE

GNN architecture components	Candidate operations
GNN layers	1, 2, 3, 4, 5, 6, 7, 8
Convolution function	$GCNConv$, $GATConv$, $GraphConv$, $GeneralConv$, $MFConv$, $LEConv$, $SAGEConv$
Bipartite convolution	$BiGCNConv$, $BiGATConv$, $BiGraphConv$, $BiGeneralConv$, $BiMFConv$, $BiLEConv$, $BiSAGEConv$
Local pooling	$TopKPool$, $SAGPool$, $PANPool$
Global pooling	$GlobalMaxPool$, $GlobalMeanPool$, $GlobalSumPool$

The first column means the name of the architecture component. The second column means the candidate operations of the corresponding architecture component, which is selected by revisiting various handcrafted GNN architectures for DDI prediction.

Bipartite Convolution: The bipartite graph stores the interaction information between a drug pair. In the DDI prediction task, the interactive information between a drug pair will provide valuable information for learning the drug substructures [12], [13]. AutoDDI performs bipartite graph convolution to capture the interactive information between a drug pair. Based on the Pytorch-Geometric library, we implement bipartite convolution. The candidate operations of bipartite convolution are as follows: $BiGCNConv$, $BiGATConv$, $BiGraphConv$, $BiGeneralConv$, $BiMFConv$, $BiLEConv$, and $BiSAGEConv$.

Local Pooling: Local pooling obtains the coarse graph in each layer, and then the GNN architecture can aggregate messages on the coarse graph to retain hierarchical information. Specifically, the local pooling operation first calculate a node score matrix of the origin graph based on the score function, and then choose nodes with highest score to generate a new coarse graph. Different local pooling operations calculate the score of nodes based on the different score functions. We add three local pooling operations to the search space: $TopKPool$ [48], $SAGPool$ [49], and $PANPool$ [50].

Global Pooling: The global pooling function is effective in graph classification tasks [15] such as DDI prediction. The global pooling function is an important way to transform the representations of all nodes in a given graph into a high-order graph-level representation. The global pooling function can reduce the complexity of the GNN architecture, prevent overfitting, and improve the generalization ability of the GNN architecture. In this work, we provide three global pooling methods without parameters to generate the graph-level representation vector: $GlobalMaxPool$, $GlobalMeanPool$, and $GlobalSumPool$.

In short, the drug–drug interaction prediction search space is defined, and the detailed candidate operations of all GNN architecture components are shown in Table I.

2) Reinforcement Learning Search Algorithm: To efficiently identify the optimal GNN architecture for the drug–drug interaction prediction, as shown in (1). We use a reinforcement learning search algorithm to achieve this goal. Reinforcement learning uses and updates the controller network to predict the excellent architecture. Specifically, a policy gradient method is used to update parameters θ of the controller network, where a moving average on feedback is applied to generate final reward feedback to reduce variance [51], as follows:

$$\nabla_\theta \mathbb{E}_{\mathbb{P}(m_{1:T};\theta)}[R(m(w,D_v))] = \sum_{t=1}^{T} \mathbb{E}_{\mathbb{P}(m_{1:T};\theta)}[\nabla_\theta \log \mathbb{P}(m_t|m_{t-1:1};\theta) R(m(w,D_v))] \quad (2)$$

where $m_{1:T}$ represents a sequence description from all architecture components generated by the controller network, where each value in the sequence represents a candidate operation for an architecture component. m represents the child GNN architecture constructed by the sequence description. w is the parameters of the child GNN architecture m, where m is trained on the training set D_t. R is the classification accuracy of the child GNN architecture m and is used as reward feedback to guide reinforcement learning to update parameters θ of the controller network. The controller network will maximize the expected accuracy of the child GNN architecture on the validation set D_v as training time increases. When the reinforcement learning search is completed, AutoDDI obtains the optimal candidate operation corresponding to each architecture component to construct the optimal GNN architecture.

The complete process of automated GNN architecture design is described in Algorithm 1.

D. Drug–Drug Interaction Prediction

DDI prediction can be converted to a binary graph classification problem. As with previous works [12], [13], we follow a recognized strategy [52] for generating negative samples. Specifically, the existing DDI triplets (d_i, r, d_j) are regarded as positive samples, while negative samples in DDI prediction are obtained by corrupting d_i or d_j, that is, by replacing d_i or d_j to generate negative samples. The sample rate for positive and negative samples in this work is 1:1. In the stage of drug–drug interaction prediction, AutoDDI constructs the optimal GNN architecture based on the optimal GNN architecture to encode the molecular structure graphs $G_i(X_i, A_i)$ and $G_j(X_j, A_j)$ of a drug pair, thus obtaining the graph representations of a drug pair. Then, AutoDDI uses MLP to predict the DDI probability between two drugs based on the graph representations of a drug pair. The cross-entropy loss function $\mathcal{L}$ provided in (3) is used to calculate the loss of the optimal GNN architecture, as shown below:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{s=(d_i,r,d_j)}^{\mathcal{D}} (\log(p_s) + \log(1-p'_s)) \quad (3)$$

$$p_s = \sigma(\mathrm{MLP}(\boldsymbol{h}_i \boldsymbol{M}_r \boldsymbol{h}_j)) \quad (4)$$

where p_s and p'_s represent the DDI probability of positive samples and negative samples, respectively; $\mathcal{D}$ denotes all DDI triplets in the dataset; $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ represent graph-level representation vectors of drug d_i and drug d_j, respectively; $\boldsymbol{M}_r$ denotes learnable parameter matrix for the interaction type r; σ represents the $Sigmoid$ activate function.

Algorithm 1: Automated GNN Architecture Design.

Input: the drug–drug interaction prediction search space M, the training set D_t, the validation set D_v, the training epoch of the controller network N, the number of predicted GNN architectures B.

Output: the optimal GNN architecture m^*.

// **Step 1: Train the Controller Network in Reinforcement Learning**
$controller_\theta \Leftarrow initialize\_controller(M)$
while not reached maximum epochs N **do**
  // **Sample the Candidate Operation from Each Architecture Component**
  $architectures \Leftarrow \emptyset$
  for $c \Leftarrow 0$ to $all\_component\_num(M)$ **do**
    $component \Leftarrow controller\_sample(controller_\theta, M)$
    $architectures.append(component)$
  end for
  $archi_w \Leftarrow build\_GNN\_archi(architectures)$
  $archi_w \Leftarrow train\_GNN\_archi(archi_w, D_t)$
  // **Generate Moving Average Feedback on** D_v
  $feedback \Leftarrow generate\_feedback(archi_w, D_v)$
  // **Update Parameters of the Controller Network**
  $controller_\theta \Leftarrow update\_controller(controller_\theta, feedback)$
end while
// **Step 2: Trained Controller Network Generate Promising GNN Architectures**
$archi_{pred} \Leftarrow controller\_predict(controller_\theta, M, B)$
$promising\_gnns \Leftarrow \emptyset$
for $archi_w \Leftarrow archi_{pred}$ **do**
  $archi_w \Leftarrow train\_GNN\_archi(archi_w, D_t)$
  $gnn, accuracy \Leftarrow generate\_accuracy(archi_w, D_v)$
  $promising\_gnns.append([gnn, accuracy])$
end for
// **Step 3: Select the Optimal GNN Architecture** m^* **from** $promising\_gnns$ **Based on Validation Accuracy**
$m^* \Leftarrow get\_optimal\_archi(promising\_gnns)$
return m^*

IV. Experiment

A. Datasets and Preprocessing

We use the DrugBank and Twosides datasets, which are widely used in the DDI prediction task, to evaluate AutoDDI. In these datasets, the molecular structure of drugs is described in SMILES format. We use the RDKit[1] python library to generate the molecular structure graph (i.e., the feature representation) of a single drug from SMILES. We combine two graphs of a drug pair to construct a bipartite graph in this work, where the nodes are the atoms of one drug that do not intersect with those of the other drug, and connect the nodes/atoms of the two drugs one by one to form edges. Thus, the bipartite graph stores the interaction information between a drug pair, which will provide valuable information for learning the drug substructures [12], [13] in the DDI prediction task. The node feature representation of one drug is generated by jointly aggregating the features of all atoms of the other drug, which helps to improve the model performance.

As with previous works [12], [13], we follow a recognized strategy [52] for generating negative samples. Specifically, the existing DDI triplets (d_i, r, d_j) are regarded as positive samples, while negative samples in DDI prediction are obtained by corrupting d_i or d_j, that is, by replacing d_i or d_j to generate negative samples. The sample rate for positive and negative samples in this work is 1:1.

DrugBank: The DrugBank dataset contains 191808 DDI triplets with 1706 drugs and 86 interaction types [53]. A drug pair in the DrugBank dataset only holds one interaction type, which indicates how one drug participates in the metabolism of another drug. For the transductive setting, the ratio of training, validation, and test sets is set to 6:2:2, which follows previous DDI prediction works [12], [13]. For the inductive setting, the ratio of training, validation, and test sets also follow previous works [12], [13], and more detailed content can be seen in the *Section IV-E*.

Twosides: The Twosides dataset contains 4649441 DDI triplets with 645 drugs and 1317 interaction types [2]. A drug pair in the Twosides dataset can hold multiple interaction types, which differs from the DrugBank dataset. If the drug pairs with the same interaction type occur less than 500 times in the dataset, then remove that interaction type, thus keeping 4576287 drug pairs with 963 interaction types. This dataset preprocessing approach follows the same criterion [54] of previous research. For the transductive setting, the ratio of training, validation, and test sets for the Twosides dataset is also set to 6:2:2, following previous works [12], [13].

B. Experimental Settings

The experimental settings for this work consist of two parts, including automated graph neural network design settings and GNN test settings.

Automated Graph Neural Network Design Settings: In the process of automated GNN design, the corresponding parameter settings follow the default settings of the previous GNAS research [32]. The training epoch of the controller network N is set to 100. The optimizer for the model parameters of the controller network is $Adam$, with a learning rate of 3.5e-4. After the controller network is trained for one epoch on the training set D_t, the model parameters of the controller network are updated using the feedback generated on the validation set D_v, where a moving average on feedback is applied to generate final feedback for one epoch. The previous works [12], [13] perform the DDI prediction based on 3-fold cross-validation, where the ratio of training, validation, and test sets is set to 6:2:2. In this work, we perform automatic GNN architecture design based on the 0-th fold data. The number of predicted GNN architectures B is set to 100. In addition, considering the computational cost of GNAS,

[1]RDKit: Open-source cheminformatics. https://www.rdkit.org

the training epoch of each GNN architecture sampled/predicted by the controller network is set to 5 in this work.

GNN Test Settings: When the reinforcement learning search is completed, AutoDDI will obtain the optimal candidate operations for each GNN architecture component, which will be used to build the optimal GNN architecture. Then, AutoDDI uses the optimal GNN architecture for conducting complete model testing, where the optimal GNN architecture does not require fine-tuning of hyperparameters as the searched GNN architecture has already adapted to hyperparameters. In the model testing process, the optimal GNN architecture performs the DDI prediction task with 3-fold cross-validation. In each fold, the learning parameters of the optimal GNN architecture will be re-initialized with $Xavier$ initialization, and the data split ratio of training/validation/test will be consistent with the automated GNN design. The parameter settings of the optimal GNN architecture are consistent with the automated GNN design and follow the previous DDI prediction works [12], [13]. The detailed parameter settings are as follows: using the $Adam$ optimizer, train the optimal GNN architecture with 200 epochs on the DrugBank dataset and 120 epochs on the Twosides dataset; the learning rate of the transductive setting is set to 0.01, and the inductive setting is set to 0.001; train the optimal GNN architecture on mini-batches with 512 DDIs.

C. Evaluation Indicators

To better present the performance of AutoDDI, we evaluate the optimal GNN architecture identified by AutoDDI with four metrics widely used in the DDI prediction task. The metrics include ACC (accuracy), AUROC (area under the receiver operating characteristic curve), AP (average precision), and F1 (F1-score).

$$
\begin{aligned}
ACC &= \frac{TP+TN}{TP+TN+FP+FN} \\
AUROC &= \sum_{i=2}^{n} \frac{(FPR_i - FPR_{i-1}) \times (TPR_i + TPR_{i-1})}{2} \\
FPR &= \frac{FP}{FP+TN} \\
TPR &= \frac{TP}{TP+FN} \\
AP &= \sum_{i=1}^{n} (R_i - R_{i-1}) \cdot P_i, \quad R_0 = 0 \\
P &= \frac{TP}{TP+FP} \\
R &= \frac{TP}{TP+FN} \\
F1 &= \frac{2 \times P \times R}{P+R}
\end{aligned}
\tag{5}
$$

where TP is the number of positive samples of correct prediction (i.e., true positive), TN is the number of negative samples of correct prediction (i.e., true negative), FP is the number of positive samples of incorrect prediction (i.e., false positive) and FN is the number of negative samples of incorrect prediction (i.e., false negative). P is precision and R is recall. ACC represents the ratio of correct prediction in all samples. $AUROC$ represents the ability of the model to correctly sort positive and negative samples. AP means the performance of the model at different recall levels. $F1$ means the harmonic mean of precision and recall.

D. Baseline Methods

To demonstrate the superiority of our AutoDDI method, we evaluated the performance of AutoDDI on both transductive and inductive settings. We compared AutoDDI with handcrafted GNN architectures.

MR-GNN: MR-GNN [9] uses GNN to obtain node representations based on message passing, then captures substructure representations of different sizes for each drug. These representations would be fed into a recurrent neural network to perform DDI prediction.

MHCADDI: In MHCADDI [10], the representation of each drug is updated based on the joint drug–drug information in the message passing process, which is obtained by integrating the interaction across drugs using a co-attention mechanism.

SSI-DDI: SSI-DDI [11] uses the hidden representations of each node as drug substructures, and then calculates the interactions between these drug substructures to conduct DDI prediction.

GMPNN-CS: GMPNN-CS [12] captures drug substructures with different sizes, and then weights the interactions between drug substructures of a drug pair for final DDI prediction.

SA-DDI: SA-DDI [14] uses a substructure-aware GNN, which extracts size-adaptive drug substructures for DDI prediction based on a novel substructure attention mechanism.

DSN-DDI: In DSN-DDI [13], the drug–drug interaction is integrated to capture the substructures of each drug, and the drug–drug interaction is abstracted as a bipartite graph.

E. Comparison Results

We compared the optimal GNN architecture identified by AutoDDI with the handcrafted GNN architectures on both transductive and inductive settings.

1) Transductive Setting: In the transductive setting, two widely used standard benchmarks, DrugBank and Twosides, are used for performance comparison. In each dataset, the molecular structure graph of each drug is obtained from SMILES using the RDKit tool, and the input feature dimension of each node in the molecular structure graph is 55 [13]. Similar to previous studies [12], [13], under this setting, the drugs used in the training set also exist in the test set. The process of splitting the dataset also follows previous studies [12], [13] to maintain the fairness of performance comparison. The ratio of positive and negative samples is 1:1, and the data split is based on classes; The data split ratio of training, validation, and test set for each class is 6:2:2. All methods perform 3-fold cross-validation with the above data split ratio.

Experiment results are presented with the means and standard deviations by 3-fold cross-validation, as shown in Table II. The

TABLE II
PERFORMANCE COMPARISON BETWEEN AUTODDI AND STATE-OF-THE-ART BASELINES ON THE TRANSDUCTIVE SETTING

Methods	DrugBank				Twosides			
	ACC	AUROC	AP	F1	ACC	AUROC	AP	F1
MR-GNN	96.04 ± 0.05	98.87 ± 0.04	98.57 ± 0.06	96.10 ± 0.05	76.23 ± 0.23	85.00 ± 0.22	84.32 ± 0.35	77.88 ± 0.35
MHCADDI	83.80 ± 0.27	91.16 ± 0.31	89.26 ± 0.37	85.06 ± 0.31	-	-	-	-
SSI-DDI	96.33 ± 0.09	98.95 ± 0.08	98.57 ± 0.14	96.38 ± 0.09	78.20 ± 0.14	85.85 ± 0.13	82.71 ± 0.14	79.81 ± 0.16
GMPNN-CS	95.30 ± 0.05	98.46 ± 0.01	97.94 ± 0.02	95.39 ± 0.05	82.83 ± 0.14	90.07 ± 0.12	87.24 ± 0.12	84.08 ± 0.14
SA-DDI	96.23 ± 0.01	98.80 ± 0.02	98.36 ± 0.04	96.29 ± 0.09	87.45 ± 0.03	93.17 ± 0.04	90.51 ± 0.08	88.35 ± 0.04
DSN-DDI	96.94 ± 0.02	99.47 ± 0.01	99.37 ± 0.02	96.93 ± 0.02	98.83 ± 0.04	99.90 ± 0.01	99.89 ± 0.01	98.83 ± 0.04
AutoDDI	**97.63 ± 0.04**	**99.53 ± 0.04**	**99.52 ± 0.05**	**97.63 ± 0.04**	**98.93 ± 0.02**	**99.92 ± 0.01**	**99.90 ± 0.01**	**98.93 ± 0.02**

The baselines are all handcrafted gnn architectures. The performance of each method is presented in the format of mean ± STD. In each metric, all results are shown in percentages and the bold number represents the best performance. '-' represents that the performance is unavailable in published work.

TABLE III
PERFORMANCE COMPARISON OF AUTODDI AND STATE-OF-THE-ART BASELINES ON THE DRUGBANK DATASET UNDER THE INDUCTIVE SETTING

Methods	S1 partition (new drug, new drug)				S2 partition (new drug, existing drug)			
	ACC	AUROC	AP	F1	ACC	AUROC	AP	F1
MR-GNN	62.63 ± 0.77	70.92 ± 0.84	73.01 ± 1.23	45.81 ± 2.51	74.67 ± 0.33	83.15 ± 0.60	83.81 ± 0.69	69.88 ± 0.86
MHCADDI	66.50 ± 0.62	72.53 ± 0.92	71.06 ± 1.61	67.21 ± 0.59	70.58 ± 0.94	77.84 ± 1.08	76.16 ± 1.45	72.74 ± 0.65
SSI-DDI	65.40 ± 1.30	73.43 ± 1.81	75.03 ± 1.42	54.12 ± 3.46	76.38 ± 0.92	84.23 ± 1.05	84.94 ± 0.76	73.54 ± 1.50
GMPNN-CS	68.57 ± 0.30	74.96 ± 0.40	75.44 ± 0.50	65.32 ± 0.23	77.72 ± 0.30	84.84 ± 0.15	84.87 ± 0.40	78.29 ± 0.16
SA-DDI	67.15 ± 0.88	73.62 ± 1.25	73.39 ± 1.40	63.40 ± 1.53	75.55 ± 1.12	82.95 ± 1.05	84.11 ± 0.92	71.94 ± 1.57
DSN-DDI	73.42 ± 1.29	81.79 ± 1.12	81.82 ± 1.48	70.34 ± 0.98	81.92 ± 1.20	91.01 ± 0.76	91.09 ± 0.93	80.18 ± 1.49
AutoDDI	**75.39 ± 1.56**	**83.65 ± 1.47**	**83.64 ± 1.61**	**73.21 ± 1.14**	**84.62 ± 1.13**	**92.18 ± 1.20**	**91.91 ± 1.17**	**84.24 ± 1.24**

The baselines are all handcrafted gnn architectures. The performance of each method is presented in the format of mean ± STD. In each metric, all results are shown in percentages and the bold number represents the best performance. In S1 partition set, two drugs in each sample are all new drugs. In S2 partition set, two drugs in each sample have one new drug and one existing drug.

experiment results show that our AutoDDI method achieves the best performance on all evaluation metrics compared to state-of-the-art handcrafted GNN architectures, whether the DrugBank or Twosides datasets. Specifically, although state-of-the-art handcrafted GNN architectures have achieved high accuracy for the DDI prediction task, our AutoDDI method still achieved improvement on all four metrics. For example, AutoDDI achieved 0.69% improvement on ACC over the best handcrafted GNN architecture, DSN-DDI, when evaluated on DrugBank. Furthermore, the AUROC reaches 99.53% on DrugBank and 99.92% on Twosides, respectively, which indicates the optimal GNN architecture designed by AutoDDI can perform DDI prediction perfectly under the transductive setting. Therefore, these experiment results indicate that AutoDDI can effectively and automatically identify the optimal GNN architecture to capture drug substructure for DDI prediction.

2) Inductive Setting: The inductive setting is more challenging compared to the transductive setting, as the dataset of the inductive setting is split based on drugs, and there are no overlapping drugs between training and test samples. This partition scheme will lead to a situation where a new drug in the test set has no known prior drug interactions, which is called cold-start scenario. This cold-start scenario is an exceptionally challenging trial, where the prior knowledge of new drugs in the test set cannot be learned during training. Hence, it can effectively test the generalization ability of the model to new drugs. The partition scheme that splits the dataset under the inductive setting follows existing literature [11], [12], [13]. Specifically, all methods are performed with 3-fold cross-validation. In each fold, 20% of the drugs are randomly selected as new drugs, while the remaining drugs are seen as existing drugs. The dataset is divided into three sets types: training set, S1 partition set, and S2 partition set. In the training set, two drugs in each sample are all existing drugs. In S1 partition set, two drugs in each sample are all new drugs. In S2 partition set, two drugs in each sample have one new drug and one existing drug.

Experiment results on the DrugBank dataset are presented with the means and standard deviations by 3-fold cross-validation, as shown in Table III. We can observe a significant decrease in the performance of all methods under the inductive setting compared to the transductive setting, indicating that predicting DDI for new drugs is much more difficult. This is because the chemical structure between the existing drugs in the training set and the new drugs in the test set shows a significant difference [11], [12]. In addition, compared to state-of-the-art handcrafted GNN architectures, our AutoDDI method also achieves the best performance on all evaluation metrics. Specifically, AutoDDI achieved relative improvements of 2.87% and 4.06% of F1 on S1 and S2 partition sets over the best handcrafted GNN architecture, DSN-DDI. These results again indicate that AutoDDI can effectively and automatically identify the optimal GNN architecture to capture drug substructure for DDI prediction.

Briefly, whether under transductive or inductive settings, our proposed AutoDDI method can automatically design GNN architectures and achieve performance improvement. Considering the novel search space designed for drug datasets, we believe that AutoDDI, as a unified framework, can provide a valuable tool for the drug discovery field.

TABLE IV
ABLATION EXPERIMENT OF AUTODDI WITH DIFFERENT SEARCH SPACES

Fixed component	Variant	ACC	AUROC	AP	F1
GNN layers	AutoDDI (two layers)	93.75 ± 0.36	97.78 ± 0.19	97.34 ± 0.24	93.83 ± 0.34
Convolution function	AutoDDI (GCNConv)	96.34 ± 0.09	99.24 ± 0.05	99.11 ± 0.09	96.32 ± 0.09
Bipartite convolution	AutoDDI (BiGATConv)	96.90 ± 0.13	99.37 ± 0.08	99.32 ± 0.11	96.89 ± 0.12
Local pooling	AutoDDI (remove local)	95.86 ± 0.26	99.10 ± 0.09	98.97 ± 0.13	95.85 ± 0.28
Global pooling	AutoDDI (GlobalSumPool)	97.16 ± 0.10	99.38 ± 0.04	99.26 ± 0.05	97.16 ± 0.10
PAS	AutoDDI (PAS)	94.07 ± 0.34	98.01 ± 0.15	97.85 ± 0.20	94.11 ± 0.32
None	AutoDDI (all)	**97.63 ± 0.04**	**99.53 ± 0.04**	**99.52 ± 0.05**	**97.63 ± 0.04**

This ablation experiment is performed on DrugBank dataset under the transductive setting. In each metric, all results are shown in percentages and the bold number represents the best performance. The first column means the corresponding architecture component we try to explore. The second column means that we fix each architecture component into one candidate operation. In particular, autoddi (PAS) denotes that autoddi uses the search space of the PAS method [15] to automatically design GNN architectures.

F. Ablation Study

We further perform comprehensive ablation studies to prove the effectiveness of our proposed AutoDDI method. In the ablation studies, the training and test settings for all variants remain the same as before.

1) Ablation Study on the Search Space: We first explore the influence of GNN architecture components on performance in the search space. We conduct an ablation study on the search space, and the ablation results are shown in Table IV.

To evaluate how the GNN layers architecture component affects the performance, we fix the number of GNN layers into two layers and then only search for other architecture components, which is represented as the variant of AutoDDI (two layers). We fix the number of GNN layers as two because many papers [41], [42], [47] observe that shallow GNNs, such as two-layer GNN, can achieve better performance than deep GNNs. The performance of AutoDDI (two layers) is significantly lower than that of AutoDDI (all), indicating that sufficient GNN layers help to improve the predictive ability of GNN architectures in DDI prediction.

To explore the effectiveness of convolution operations for AutoDDI, we fix the convolution function component into GCNConv and then only search for other architecture components, which is represented as the variant of AutoDDI (GCNConv). The GCNConv operation is the widely used convolution component in GNN-based applications [41], [47]. Compared with AutoDDI (all), the performance of AutoDDI (GCNConv) slightly decreases, indicating that the GCNConv operation has good generalization ability for different tasks. Meanwhile, the results show that AutoDDI can automatically identify more effective convolution operations to improve the performance of DDI prediction.

The bipartite graph stores the interaction information between two drugs. To explore the effect of the bipartite graph on AutoDDI performance, we fix the bipartite convolution component into BiGATConv and then only search for other architecture components, which is represented as the variant of AutoDDI (BiGATConv). The BiGATConv operation is the most widely used in the DDI prediction task [11], [12], [13]. The performance of AutoDDI (BiGATConv) is slightly lower than that of AutoDDI (all), indicating that the BiGATConv operation performs well in the DDI prediction task. Meanwhile, the results show that AutoDDI can automatically identify more effective bipartite convolution operations to improve the performance of DDI prediction.

By the local pooling operation, the model can extract information from different levels of input features, forming a hierarchical feature representation. This helps the model better understand the structure of input features, improving its generalization ability and performance. To explore the effect of hierarchical feature representation on AutoDDI performance, we create a variant of AutoDDI (remove local), in which AutoDDI deletes the local pooling component from the DDI prediction search space, but AutoDDI still keeps the search for other architecture components. AutoDDI (remove local) shows a drop in performance compared to AutoDDI (all), indicating that the local pooling component plays an important role in DDI prediction.

To explore the effectiveness of global pooling operations for AutoDDI, we fix the global pooling component into GlobalSumPool and then only search for other architecture components, which is represented as the variant of AutoDDI (GlobalSumPool). The GlobalSumPool operation is the widely used global pooling operation in the graph classification task [15], [16], [17]. Compared with AutoDDI (all), the performance of AutoDDI (GlobalSumPool) is almost without loss. This shows that GlobalSumPool, as the typical global pooling operation, has excellent feature extraction capabilities. At the same time, the results show that AutoDDI can automatically identify more effective global pooling operations to improve the performance of DDI prediction.

Additionally, to demonstrate the novelty of the search space proposed by our AutoDDI method, we compare it with the search space designed in prior works. AutoDDI uses the search space of the PAS method [15] to automatically design GNN architectures, which is represented as the variant of AutoDDI (PAS). The PAS method automatically designs convolution functions, local pooling operations, and global pooling operations. However, PAS did not consider the influence of the number of GNN layers and did not design the operation for the bipartite graph in the drug dataset. The performance of AutoDDI (PAS) is significantly lower than that of AutoDDI (all), indicating the importance of the number of GNN layers and the interaction information of the bipartite graph. This observation also demonstrates the novelty of the search space proposed by our AutoDDI method.

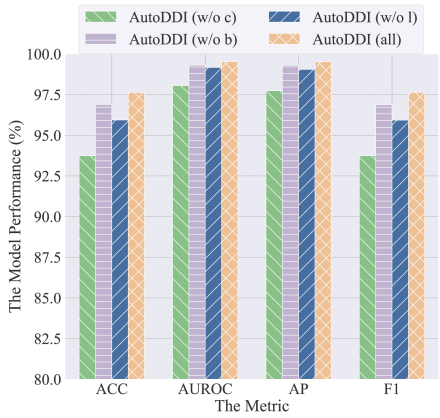

Fig. 3. Ablation experiment of the optimal GNN architecture identified by AutoDDI. This ablation experiment is performed on the DrugBank dataset under the transductive setting. We show the model performance of the optimal GNN architecture by different metrics. Each variant of AutoDDI denotes the corresponding architecture component removed from the optimal GNN architecture.

2) Ablation Study on the Optimal GNN Architecture: We then explore the impact of several operations on performance in the optimal GNN architecture identified by AutoDDI. We conduct an ablation study on the optimal GNN architecture, and the ablation results are shown in Fig. 3. The variant of AutoDDI (w/o c) represents the optimal GNN architecture that deletes the convolution function. The variant of AutoDDI (w/o b) denotes the optimal GNN architecture that does not use bipartite convolution. The variant of AutoDDI (w/o l) means the optimal GNN architecture that deletes the local pooling. The variant of AutoDDI (all) represents the complete optimal GNN architecture. The experimental results show that the convolution operation significantly affects the performance of the optimal GNN architecture, and the bipartite convolution operation has a smaller impact on the performance of the optimal GNN architecture. This indicates that the features of the two drugs themselves are the key factors determining DDI prediction, and the interaction information between the two drugs is also helpful for DDI prediction. In addition, the ablation results of the variant of AutoDDI (w/o l) again show that the hierarchical feature representation generated by local pooling operations helps to improve the generalization ability of the optimal GNN architecture.

G. Case Study

In this section, we carry out a case study to verify that our proposed AutoDDI method is able to effectively capture substructures that cause DDIs to occur. The case study experiment is conducted on the DrugBank dataset under the transductive setting. For a demonstration, we visualize the drug substructure for DDIs between the *dicoumarol* drug and the other five drugs, as shown in Fig. 4. We can see that these examples are confidently DDIs with high prediction probabilities. AutoDDI captures the drug substructures of barbituric acid for *amobarbital*, *pentobarbital*, *secobarbital*, *methylphenobarbital*, and *primidone* drugs, which is in accord with the fact that the drug with a barbituric acid substructure enhances the activity of

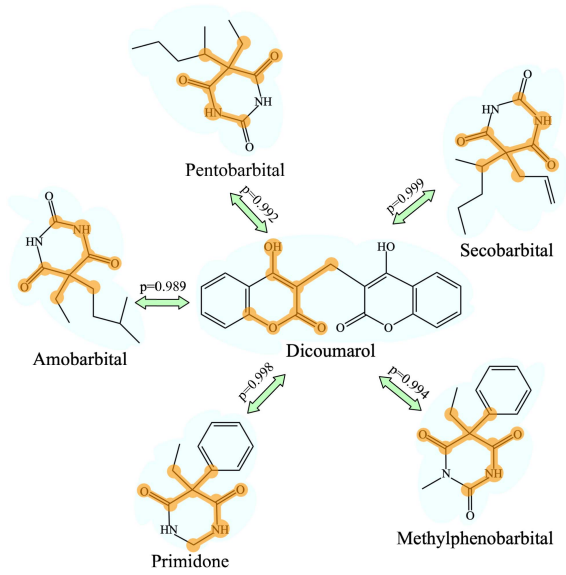

Fig. 4. Visualization of drug substructures for DDIs. The DDIs between the *dicoumarol* drug and the other five drugs are displayed, where the predicted probabilities of DDIs are marked and the drug substructures captured by AutoDDI are shown in orange. Visualization results indicate that classical two-layer GNN architectures are insufficient to capture the drug substructure. This case study experiment is conducted on the DrugBank dataset under the transductive setting.

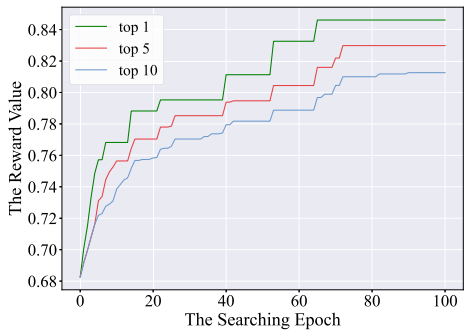

Fig. 5. Convergence analysis of reinforcement learning searching process in our AutoDDI method. The x-axis means the searching epoch of reinforcement learning. The y-axis means the reward value of the best GNN architectures during search, where green represents the best reward, red represents the average reward of top 5, and blue represents the average reward of top 10. Each GNN architecture is trained for 5 epochs on training set and its validation accuracy is used as a reward value. This convergence analysis experiment is performed on the DrugBank dataset under the transductive setting.

the liver microsome, thereby decreasing the curative effect of the *dicoumarol* drug [55]. We also demonstrate the rationality of capturing the drug substructure by stacking GNN layers because the classical two-layer GNN architecture is insufficient to capture the barbituric acid substructure. To sum up, the working mechanism of GNNs in drug discovery scenarios can be better understood by exploring the drug substructure for DDIs.

H. Model Interpretability

In this section, we list the optimal GNN architecture designed by AutoDDI for each drug dataset, as shown in Table V. We provide candidate operation information corresponding to each

TABLE V
OPTIMAL GNN ARCHITECTURE DESIGNED BY AUTODDI FOR EACH DRUG DATASET

GNN architecture components	Transductive setting		Inductive setting
	DrugBank	Twosides	DrugBank
GNN layers	*6*	*5*	*8*
Convolution function	*MFConv*	*GATConv*	*GCNConv*
Bipartite convolution	*GCNConv*	*GATConv*	*GATConv*
Local pooling	*SAGPool*	*TopKPool*	*SAGPool*
Global pooling	*GlobalMeanPool*	*GlobalSumPool*	*GlobalMeanPool*

Each row represents the corresponding GNN architecture component in drug–drug interaction prediction search space. Each column represents a drug dataset, in which the optimal candidate operations of each architecture component in the optimal GNN architecture are shown.

GNN architecture component in the optimal GNN architecture. We can see that the candidate operation of each GNN architecture component varies across different datasets for all optimal GNN architectures. For example, the candidate operation of both the GNN layers and the convolution function vary across different datasets. These results show that AutoDDI can obtain the data-specific optimal GNN architecture for each drug dataset.

We demonstrate the rationality of the optimal GNN architecture designed by AutoDDI. The GNN layers in architectures designed by AutoDDI are deep, because the molecule graph of the drug is large, thereby classical two-layer GNN architectures are insufficient to capture the information from long-distant neighbors, which is a key factor for capturing the drug substructure. For example, visualization results of Fig. 4 show that classical two-layer GNN architectures are insufficient to capture the barbituric acid drug substructure (colored by orange). In addition, the DrugBank and Twosides datasets respectively tend to choose *GCNConv* and *GATConv* to aggregate the information from neighbors. A drug pair in the DrugBank dataset only holds one interaction type, but a drug pair in the Twosides dataset can hold multiple interaction types. *GCNConv* only has a set of convolutional kernels with shared parameters, and can better capture homogeneous (only one interaction type on the DrugBank dataset) neighbor information. *GATConv* introduces multiple sets of convolutional kernels with different attention coefficients by the attention mechanism, and can better capture heterogeneous (multiple interaction types on the Twosides dataset) neighbor information.

I. Convergence Analysis

To evaluate the effectiveness of AutoDDI in automatically designing GNN architectures, we conduct a convergence analysis experiment on the reinforcement learning searching process. The convergence analysis experiment is performed on the DrugBank dataset under the transductive setting. Here, each GNN architecture is trained for 5 epochs on training set and its validation accuracy is used as a reward value. We obtain the reward value of the best GNN architectures during each searching epoch. The higher reward value derived from the best GNN architecture reflects the ability of the reinforcement learning search algorithm. Thus, we present the reward value of the best GNN architecture to show the convergence of reinforcement learning searching process in our AutoDDI method, as shown by green line in Fig. 5. We also present the average reward of top 5 (red line) and top 10 (blue line) GNN architectures during each searching epoch. The visualization results show that as the searching process progresses, the search direction based on reinforcement learning will gradually converge to the area that includes promising GNN architectures with higher reward values.

V. CONCLUSION AND FUTURE DIRECTION

In this work, we propose a novel automated graph neural network method called AutoDDI for the drug–drug interaction prediction task. To this end, we design an effective search space by revisiting various handcrafted GNN architectures for drug–drug interaction prediction. Based on the search space, AutoDDI uses the reinforcement learning search algorithm to automatically search the optimal GNN architecture to capture drug substructure for drug–drug interaction prediction. Experimental results demonstrate that our AutoDDI method can achieve the best performance than state-of-the-art handcrafted GNN architectures on two real-world datasets, DrugBank and Twosides. The results of the ablation study show that the number of GNN layers can significantly affect model performance, indicating the importance and rationality of automatically adapting GNN depth for each drug dataset. Moreover, the ablation study demonstrates that the interaction information of drug pairs captured by bipartite convolution architecture component can help improve the model performance. The visual interpretation results of the case study provide more insights into how AutoDDI effectively captures drug substructure to predict drug–drug interaction.

For future direction, we will incorporate domain knowledge for heuristic search, such as using knowledge extracted by knowledge distillation as part of reward feedback to guide search algorithm iterations. Besides, we plan to introduce other deep learning techniques for AutoDDI, such as using a differentiable architecture search algorithm to convert the discrete search process into the continuous search process to improve search efficiency.

REFERENCES

[1] S. Jain, E. Chouzenoux, K. Kumar, and A. Majumdar, "Graph regularized probabilistic matrix factorization for drug-drug interactions prediction," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 5, pp. 2565–2574, May 2023.

[2] N. P. Tatonetti, P. P. Ye, R. Daneshjou, and R. B. Altman, "Data-driven prediction of drug effects and interactions," *Sci. Transl. Med.*, vol. 4, 2012, Art. no. 125ra31.

[3] X. Sun, S. Vilar, and N. P. Tatonetti, "High-throughput methods for combinatorial drug discovery," *Sci. Transl. Med.*, vol. 5, 2013, Art. no. 205rv1.

[4] J.-Y. Shi, K.-T. Mao, H. Yu, and S.-M. Yiu, "Detecting drug communities and predicting comprehensive drug–drug interactions via balance regularized semi-nonnegative matrix factorization," *J. Cheminformatics*, vol. 11, pp. 1–16, 2019.

[5] Y. Shang, L. Gao, Q. Zou, and L. Yu, "Prediction of drug-target interactions based on multi-layer network representation learning," *Neurocomputing*, vol. 434, pp. 80–89, 2021.

[6] Z.-H. Ren et al., "DeepMPF: Deep learning framework for predicting drug–target interactions based on multi-modal representation with meta-path semantic analysis," *J. Transl. Med.*, vol. 21, 2023, Art. no. 48.

[7] K. Huang, C. Xiao, T. Hoang, L. Glass, and J. Sun, "Caster: Predicting drug interactions with chemical substructure representation," in *Proc. Assoc. Adv. Artif. Intell.*, 2020, pp. 702–709.

[8] Y. Chen, T. Ma, X. Yang, J. Wang, B. Song, and X. Zeng, "MUFFIN: Multi-scale feature fusion for drug–drug interaction prediction," *Bioinf.*, vol. 37, pp. 2651–2658, 2021.

[9] N. Xu, P. Wang, L. Chen, J. Tao, and J. Zhao, "MR-GNN: Multi-resolution and dual graph neural network for predicting structured entity interactions," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3968–3974.

[10] A. Deac, Y.-H. Huang, P. Veličković, P. Liò, and J. Tang, "Drug–drug adverse effect prediction with graph co-attention," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2019, pp. 1–8.

[11] A. K. Nyamabo, H. Yu, and J.-Y. Shi, "SSI–DDI: Substructure–substructure interactions for drug–drug interaction prediction," *Brief. Bioinf.*, vol. 22, 2021, Art. no. bbab133.

[12] A. K. Nyamabo, H. Yu, Z. Liu, and J.-Y. Shi, "Drug–drug interaction prediction with learnable size-adaptive molecular substructures," *Brief. Bioinf.*, vol. 23, 2022, Art. no. bbab441.

[13] Z. Li, S. Zhu, B. Shao, X. Zeng, T. Wang, and T.-Y. Liu, "DSN-DDI: An accurate and generalized framework for drug–drug interaction prediction by dual-view representation learning," *Brief. Bioinf.*, vol. 24, 2023, Art. no. bbac597.

[14] Z. Yang, W. Zhong, Q. Lv, and C. Y.-C. Chen, "Learning size-adaptive molecular substructures for explainable drug–drug interaction prediction by substructure-aware graph neural network," *Chem. Sci.*, vol. 13, pp. 8693–8703, 2022.

[15] L. Wei, H. Zhao, Q. Yao, and Z. He, "Pooling architecture search for graph classification," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2091–2100.

[16] J. Chen, J. Gao, Y. Chen, B. M. Oloulade, T. Lyu, and Z. Li, "Auto-GNAS: A parallel graph neural architecture search framework," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 3117–3128, Nov. 2022.

[17] L. Wei, Z. He, H. Zhao, and Q. Yao, "Search to capture long-range dependency with stacking GNNs for graph classification," in *Proc. ACM World Wide Web Conf.*, 2023, pp. 588–598.

[18] P. Zhang, F. Wang, J. Hu, and R. Sorrentino, "Label propagation prediction of drug–drug interactions based on clinical side effects," *Sci. Rep.*, vol. 5, 2015, Art. no. 12339.

[19] T. Ma, C. Xiao, J. Zhou, and F. Wang, "Drug similarity integration through attentive multi-view graph auto-encoders," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3477–3483.

[20] H. Yu et al., "Predicting and understanding comprehensive drug–drug interactions via semi-nonnegative matrix factorization," *BMC Syst. Biol.*, vol. 12, pp. 101–110, 2018.

[21] W. Zhang, Y. Chen, F. Liu, F. Luo, G. Tian, and X. Li, "Predicting potential drug–drug interactions by integrating chemical, biological, phenotypic and network data," *BMC Bioinf.*, vol. 18, pp. 1–12, 2017.

[22] W. Zhang, Y. Chen, D. Li, and X. Yue, "Manifold regularized matrix factorization for drug–drug interaction prediction," *J. Biomed. Inform.*, vol. 88, pp. 90–97, 2018.

[23] H. Wang, D. Lian, Y. Zhang, L. Qin, and X. Lin, "GoGNN: Graph of graphs neural network for predicting structured entity interactions," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 1317–1323.

[24] Y.-H. Feng, S.-W. Zhang, and J.-Y. Shi, "DPDDI: A deep predictor for drug–drug interactions," *BMC Bioinf.*, vol. 21, pp. 1–15, 2020.

[25] D. Lukovnikov and A. Fischer, "Improving breadth-wise backpropagation in graph neural networks helps learning long-range dependencies," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7180–7191.

[26] Z. Wu, P. Jain, M. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, "Representing long-range context for graph neural networks with global attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 13266–13279.

[27] V. P. Dwivedi et al., "Long range graph benchmark," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 22326–22340.

[28] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. Assoc. Advance. Artif. Intell.*, 2018, pp. 3538–3545.

[29] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5453–5462.

[30] B. M. W. Harrold and R. M. Zavod, "Basic concepts in medicinal chemistry," *Drug Develop. Ind. Pharm.*, vol. 40, p. 988, 2014.

[31] J. Chen, J. Gao, Y. Chen, M. B. Oloulade, T. Lyu, and Z. Li, "GraphPAS: Parallel architecture search for graph neural networks," in *Proc. ACM Int. Conf. Special Int. Group Inf. Retrieval*, 2021, pp. 2182–2186.

[32] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, "Graph neural architecture search," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 1403–1409.

[33] M. Yoon, T. Gervet, B. Hooi, and C. Faloutsos, "Autonomous graph mining algorithm search with best speed/accuracy trade-off," in *Proc. IEEE Int. Conf. Data Mining*, 2020, pp. 751–760.

[34] Y. Li, Z. Wen, Y. Wang, and C. Xu, "One-shot graph neural architecture search with dynamic search space," in *Proc. Assoc. Adv. Artif. Intell.*, 2021, pp. 8510–8517.

[35] Y. Li and I. King, "AutoGraph: Automated graph neural network," in *Proc. Int. Conf. Neural Inf. Process.*, 2020, pp. 189–201.

[36] H. Zhao, L. Wei, and Q. Yao, "Simplifying architecture search for graph neural network," in *Proc. ACM Int. Conf. Inf. Knowl. Manage. Workshop*, 2020, pp. 1–8.

[37] Z. Huan, Y. Quanming, and T. Weiwei, "Search to aggregate neighborhood for graph neural network," in *Proc. IEEE Int. Conf. Data Eng.*, 2021, pp. 552–563.

[38] L. Wei, H. Zhao, and Z. He, "Designing the topology of graph neural networks: A novel feature fusion perspective," in *Proc. ACM World Wide Web Conf.*, 2022, pp. 1381–1391.

[39] S. Jiang and P. Balaprakash, "Graph neural network architecture search for molecular property prediction," in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 1346–1353.

[40] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," in *Proc. Int. Conf. Learn. Represent. Workshop*, 2019, pp. 1–9.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[43] C. Morris et al., "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proc. Assoc. Adv. Artif. Intell.*, 2019, pp. 4602–4609.

[44] J. You, R. Ying, and J. Leskovec, "Design space for graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17009–17021.

[45] D. K. Duvenaud et al., "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[46] E. Ranjan, S. Sanyal, and P. Talukdar, "ASAP: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proc. Assoc. Adv. Artif. Intell.*, 2020, pp. 5470–5477.

[47] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[48] H. Gao and S. Ji, "Graph u-nets," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2083–2092.

[49] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3734–3743.

[50] Z. Ma, J. Xuan, Y. G. Wang, M. Li, and P. Liò, "Path integral based convolution and pooling for graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16421–16433.

[51] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999.

[52] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. Assoc. Adv. Artif. Intell.*, 2014, pp. 1112–1119.

[53] D. S. Wishart et al., "DrugBank 5.0: A major update to the drugbank database for 2018," *Nucleic Acids Res.*, vol. 46, pp. D1074–D1082, 2018.

[54] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinf.*, vol. 34, pp. i457–i466, 2018.

[55] C. Ioannides and D. V. Parke, "Mechanism of induction of hepatic microsomal drug metabolizing enzymes by a series of barbiturates," *J. Pharm. Pharmacol.*, vol. 27, pp. 739–746, 1975.