GoGNN: Graph of Graphs Neural Network for Predicting Structured Entity Interactions

Hanchen Wang\textsuperscript{1}, Defu Lian\textsuperscript{2*}, Ying Zhang\textsuperscript{1}, Lu Qin\textsuperscript{1}, Xuemian Lin\textsuperscript{3}

\textsuperscript{1}University of Technology Sydney
\textsuperscript{2}University of Science and Technology of China
\textsuperscript{3}University of New South Wales

hanchenw.au@gmail.com, liandefu@ustc.edu.cn,
\{ying.zhang, lu.qin\}@uts.edu.au, lxue@cse.unsw.edu.au

Abstract

Entity interaction prediction is essential in many important applications such as chemistry, biology, material science, and medical science. The problem becomes quite challenging when each entity is represented by a complex structure, namely structured entity, because two types of graphs are involved: local graphs for structured entities and a global graph to capture the interactions between structured entities. We observe that existing works on structured entity interaction prediction cannot properly exploit the unique graph of graphs model. In this paper, we propose a Graph of Graphs Neural Network, namely GoGNN, which extracts the features in both structured entity graphs and the entity interaction graph in a hierarchical way. We also propose the dual-attention mechanism that enables the model to preserve the neighbor importance in both levels of graphs. Extensive experiments on real-world datasets show that GoGNN outperforms the state-of-the-art methods on two representative structured entity interaction prediction tasks: chemical-chemical interaction prediction and drug-drug interaction prediction. Our code is available at Github\textsuperscript{1}.

1 Introduction

Interactions between the structured entities like chemicals are the basis of many applications such as chemistry, biology, material science, medical science, and environmental science. For example, the knowledge of chemical interactions is a helpful guide for the toxicity prediction, new material design and pollutant removal [Xu et al., 2019]. In medical science, understanding the interaction between drugs is vital for drug discovery and side effect prediction which can save millions of lives every year [Menche et al., 2015].

One immediate way to investigate the interactions between two structured entities is to conduct experiments for them in the laboratory or clinics. However, due to the enormous number of structured entities, it is infeasible in terms of both time and resource to examine all possible interactions. Thanks to...
dict the structured entity interactions. In particular, we propose a novel model called Graph of Graphs Neural Network (GoGNN). Our model builds a graph neural network with attention-based pooling over local graphs and attention-based neighbor aggregation on the global graph such that GoGNN is able to capture broader information that enhances the performance on the prediction. Furthermore, the GNNs on both levels of graphs play synergistic effects on improving the representativeness of GoGNN. The contributions of our model can be summarized as follows:

- To the best of our knowledge, this is the first work to systematically apply the graph neural network on graph-of-graphs model, namely Graph of Graphs Neural Network (GoGNN), to the problem of structured entity interaction prediction.
- The proposed GoGNN mines the features from both local entity graphs and global interaction graph hierarchically and synergistically. We design dual attention architecture to capture the significance of the substructures in the local graphs while preserving the importance of the interactions within the global graph.
- The extensive experiments conducted on the real-life benchmark datasets show that GoGNN outperforms the state-of-the-art structured entity interaction prediction methods in two representative applications: chemical-chemical interaction prediction and drug-drug interaction prediction.

2 Related Work

In this section, we introduce the closely related works.

2.1 Structured Entities Interaction Prediction

In many real-life applications such as chemistry, biology, material science, and medical science, we need to understand the interactions between the structured entities. In recent years, a variety of techniques have been proposed for structured entity interaction prediction in some specific applications. In this paper, we focus on two representative applications: chemical-chemical interaction prediction and drug-drug interaction prediction.

Many computational methods have been proposed for these two applications. DeepCCI and DeepDDI [Kwon and Yoon, 2017; Ryu et al., 2018] utilize the conventional convolutional neural network and PCA on the chemical data. Some models are graph neural network-based. For example, Decagon [Zitnik et al., 2018] performs the GCN on drug-protein interaction graph; MR-GNN [Xu et al., 2019] proposes a model with dual graph-state LSTMs that extracts local features of molecule graphs, and MLRDA [Chu et al., 2019] utilizes graph autoencoder with a novel loss function to predict the drug-drug interactions.

2.2 Graph Neural Networks

Node-level applications. Most GNNs are designed for node-level applications such as node classification and link prediction [Kipf and Welling, 2017; Velickovic et al., 2017; Zhang and Chen, 2018; Hamilton et al., 2017; Liu et al., 2019; Lian et al., 2020b; 2020a]. They rely on the node embedding techniques like skip-gram, autoencoder and neighbor aggregation methods like GCN, GraphSAGE, etc. These methods focus on the node relations within the graph and use the low-dimension representations to preserve the structural and attribute information.

Graph-level applications. Recently, some research works on GNNs are proposed for graph-level applications such as graph classification [Zhang et al., 2018; Lee et al., 2018] and graph matching [Li et al., 2019b]. These works learn the graph representations for each graph individually or pair-wisely without considering the interactions between the graphs.

2.3 Graph of Graphs

In most real-world systems, an individual network is one component within a much larger complex multi-level network. Applying the graph theory paradigm to these networks has led to the development of the concept of “Graph of Graphs” (also known as “Network of Networks”). [D’Agostino and Scala, 2014] introduces the theoretical research development (Dong et al., 2013), applications [Ni et al., 2014] and phenomenological model [Rome et al., 2014] on the network of networks. These works enable us to understand and model the inter-dependent critical infrastructures. SEAL [Li et al., 2019a] proposed graph neural network in a hierarchical graph perspective for graph classification task. With significant differences between GoGNN and SEAL in tasks, loss functions and optimizers, GoGNN is the first work to develop graph neural network technique on graph of graphs for structured entity interaction prediction problem.

3 Preliminaries

3.1 Problem Definition

For ease of understanding of our techniques, in this paper, we focus on two representative applications of structured entities interaction prediction: chemical-chemical interaction (CCI) prediction and drug-drug interaction (DDI) prediction. In the CCI graph, there is only one type of interaction, and our goal is to estimate the reaction probability score $p_{ij}$ of given chemical pair $(G_i, G_j)$. As to DDI graph which has multiple types of interactions, we aim to estimate the occurrence probabilities $p_{ij}$ of side effect type $r$ with the given triplet $(G_i, r, G_j)$.

3.2 Input Graph of Graphs

Overall, the input interaction graph is regarded as graph-of-graphs as follows.

Molecule Graph. In both CCI and DDI prediction tasks, the local graphs are molecule graphs, each of which can be modeled as a heterogeneous graph with multiple types of nodes and edges. In particular, the molecule graph $G_M$ consists of atoms $\{a_i\}$ as nodes, and edges $\{e_{ij}\}$ where $e_{ij}$ denotes the bond between atoms $a_i$ and $a_j$. Each atom (i.e., node) $a$ is encoded as a vector $x_a$. For each bond (i.e., edge), we assign a weight to the corresponding edge depending on the type of the bond. For example, the bond between the carbon atoms in the ethylene molecule is a double bond. Therefore, the weight of the edge $e_{CC}$ between the carbon atoms is set to 2.

Interaction Graph. The global interaction graph $G_I$ is formed by the molecule graphs and the interactions between
them: \( G_I = \{ N, E_I \} \), where \( N \) denotes the node set of \( G_I \) which consists of molecule graphs \( \{ G_M \} \), and \( E_I \) denotes the interaction edges between the molecule graphs. Note that, in CCI graph, there is only one type of interaction between two nodes. In DDI graph, there are multiple types of side effects caused by the combination of two drugs. An attribute vector \( e^r \) is assigned for each edge \( e \) based on the side effect type \( r \).

4 Graph of Graphs Neural Network

In this section, we introduce our Graph of Graphs Neural Network model.

4.1 Framework of GoGNN

The framework of GoGNN is illustrated in Figure 2. GoGNN contains molecule graph neural network which takes the atom features as input and interaction graph neural network which produces the graph representation for the prediction task. The two parts of GoGNN play a synergistic effect on improving the performance. The hidden features learned by molecule-level GNN provide the interaction-level GNN a representative initial input. The feature aggregation on the interaction-level GNN promotes the ability of molecule-level GNN to find key substructure through back-propagation.

4.2 Molecule Graph Neural Network

In organic chemistry, functional groups (i.e., substructures) in molecules are responsible for the characteristic chemical reactions between these molecules. For example, the reaction between benzoic acid and ethanol in Figure 1 is the esterification between two functional groups -COOH in benzoic acid and -OH in ethanol.

The model could achieve better performance for prediction if the model can identify the functional groups in the molecules and represent the molecule with such functional groups. Therefore, we designed our molecule graph neural network with the combination of multi-resolution architecture [Xu et al., 2019] which preserves the information of multi-hop substructures and attention-based graph pooling [Lee et al., 2019; Gao and Ji, 2019a] which selects the substructures to represent the molecules.

As proved in previous work [Xu et al., 2018], one single general graph convolution layer can only aggregate the feature of the node and its immediate neighbors. To obtain features of the multi-scale substructure of the molecule graph, we apply multiple layers of graph convolution operations to the input graphs. The graph convolution operation at \( l^{th} \) layer is summarized as follows

\[
M_{(l+1)} = GCN_l(A, M_I)
\]

\[
GCN_l(A, M_I) = \sigma(D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} M_I W_l)
\]

where \( M_I \in \mathbb{R}^{n \times d} \) is the hidden feature matrix for molecule graph at \( l^{th} \) layer, \( \hat{A} = A + I \) is the adjacency matrix with self-connection for molecule graph \( G_M \), \( \hat{D} \) is the diagonal degree matrix of \( \hat{A} \) and \( \sigma(\cdot) \) denotes the activation function.

Different from MR-GNN [Xu et al., 2019] which uses dual graph-state LSTMs on the input of subgraph representations, GoGNN applies graph pooling for learning the graph representation that preserves the substructure information, in order to reduce the time and space complexity significantly. As shown in Figure 2, the self-attention graph pooling layer takes the output of each graph convolution layer as input to select the most representative substructures (functional groups) by learning the self-attention score \( s_l \in \mathbb{R}^{n \times 1} \) for molecule graph \( G_M \) with \( n \) atoms at \( l^{th} \) layer

\[
s_l = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} M_I W_{att}^l)
\]

where \( W_{att}^l \in \mathbb{R}^{d \times 1} \) is the attention weight matrix for the pooling layer to obtain the self-attention score. In order to
select the most representative substructure, the graph pooling layer calculates the attention score for each atom in the graph and finds the top-\(n\) atoms with the highest attention scores. We set a hyperparameter pooling ratio \(\gamma \in (0, 1]\) to determine the number of nodes \([\gamma n]\) that are selected to represent the molecule graph

\[
idx = \text{top}(s, [\gamma n]), \quad s_{\text{mask}} = s_{\text{idx}} \\
M_{\text{sel}} = M \odot s_{\text{mask}}
\]  

(3)

where \(\text{top}\) is the function that returns the indices of atoms with top \([\gamma n]\) attention scores as in [Gao and Ji, 2019b]; \(s_{\text{mask}} \in \{0, 1\}^{n \times 1}\) is the mask vector determined by the attention score; \(\odot\) denotes the column-wise product for masking; \(M_{\text{sel}}\) is the feature matrix of selected atoms in a molecule graph. Afterward, the readout layer, which contains mean and sum pooling, is applied on the embedding of selected atoms \(M_{\text{sel}}\) to produce the molecule graph hidden feature. After multiple graph convolutional and self-attention graph pooling layers, we get several graph hidden features. Once obtained, we concatenate the outputs of the graph pooling layers as the hidden feature vector \(x_{G,M}\) for the molecule graph. Because the hierarchical graph pooling architecture is applied, the graph representation can preserve the multi-hop substructure information effectively. Hence, GoGNN can identify the function groups which play the key roles in molecule interactions and use these functional groups to represent the molecule graph.

4.3 Interaction Graph Neural Network

Most of existing CCI and DDI prediction models train the model with the input of pair of molecule graphs, but ignore the molecule interaction graph. However, the information of interaction graph is crucial for the interaction prediction because it enables the model to capture high-order interaction relationship and enhance the model’s ability to capture the representative molecular substructures synergistically.

We have the following observations that motivate us to perform graph neural network on the interaction graph: Firstly, the type of interaction is dependent on the type of involved molecules. As mentioned in Section 4.2, esterification is the reaction between -OH in alcohols and -COOH in carboxylic acids. The neighbor aggregation of GNN can gather the neighbor information that helps to summarize the types of chemicals that interact with the selected one. Secondly, it is necessary to assign importance score to the neighbors for molecules in the interaction graph, since the chemical interactions have different significance and frequency. For example, vitamin C has two main properties: reducibility and acidity. Therefore, vitamin C cannot be prescribed with oxidizing drugs like vitamin K1 and alkaline drugs like omeprazole. In an uncommon case, vitamin C reduces the therapeutic effect of inosine because of their complex physical and chemical reactions. Therefore, we apply the graph attention network in order to preserve the frequencies of the chemical reactions and reduce the influence of biased observation of the interaction graph. As for the DDI graph with edge attributes, an edge-aggregation graph neural network is applied.

**Graph Attention Network.** The attention-based graph neural network [Veličković et al., 2017] is applied on the interaction graph without edge attributes. With the learned molecule hidden feature vector \(x_{G,m}\) and interaction graph \(G_t = \{N,E_t\}\) as input, molecule graph representations are calculated by the neighbor aggregation on the interaction graph as follows

\[
x_{G_i}^{t+1} = \frac{1}{K} \sum_{j \in \eta_{G_i}} \sigma(\{\sum_{k} \alpha_{ij}^r W_k^r x_{G_j}^t\})
\]  

(4)

where \(K\) is the number of attention heads, \(\sigma\) is a nonlinearity function, \(W_k^r\) is the weight matrix at \(k\)th attention head in \(l\)th layer and \(\eta_{G_i}\) is the set of neighbor molecule graphs of \(G_i\) in the interaction graph \(G_t\). Notation \(\alpha_{ij}^r\) is the attention coefficient between \(G_i\) and \(G_j\) which is calculated by the following equation:

\[
\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a(W x_{G_i}, \| W x_{G_j})))}{\sum_{n \in \eta_{G_i}} \exp(\text{LeakyReLU}(a(W x_{G_i}, \| W x_{G_n})))}
\]  

(5)

where \(a\) is a learnable attention weight vector and \(\|\) is the concatenation operation.

**Edge Aggregation Network.** In DDI graph, each edge has an attribute vector \(e_{ij}\) which is determined by the side effect type \(r\) of the drug combination \((G_i, G_j)\). To capture the edge attributes [Schlichtkrull et al., 2018], we propose an edge aggregation network that aggregates the neighbor information together with edge attributes:

\[
x_{G_i}^{t+1} = \sigma(W x_{G_i}, \sum_{r} \left( \sum_{j \in \eta_{G_i}} x_{G_j}^t \cdot h_{W_e}(e_{ij}^r) \right))
\]  

(6)

where \(h_{W_e}\) is the MLP layer with linear transformation matrix \(W_e\) which transforms the edge attribute vector \(e_{ij}^r \in \mathbb{R}^{h \times 1}\) into a real number \(\tau_{ij}^r \in \mathbb{R}\). In this way, GoGNN aggregates node’s neighbor information together with edge attributes. Different from Decagon [Zitnik et al., 2018] which sets side-effect-specific parameters, GoGNN shares the parameters for all types of side effects in order to improve the robustness and generalization of the model.

4.4 GoGNN Model Training

We optimize the parameters with the task-specific loss functions.

**Chemical Interaction Prediction.** Since there is no edge attribute in the graph, we regard the chemical interaction prediction as a link prediction problem. The dot product of two graph representations is used as the link probability of two graphs:

\[
p_{ij} = \sigma(x_{G_i}^T \cdot x_{G_j})
\]  

(7)

where \(\sigma\) is the activation function such as \(\text{sigmoid}\) function that ensures \(p_{ij} \in (0, 1]\). To encourage the model to assign higher probabilities to the observed edges than the random non-edges, we follow the previous study and estimate the model through negative sampling. For each positive edge pair \((G_i, G_j)\), a random negative edge \((G_i, G_m)\) is sampled by choosing a molecule graph \(G_m\) randomly. We optimize the model using the following cross-entropy loss function

\[
\mathcal{L}_{CCI} = \sum_{(G_i, G_j) \in \mathcal{E}_{CCI}} -\log(p_{ij}) - \mathbb{E}_{m \sim p_m} \log(1 - p_{im})
\]  

(8)
**Drug Interaction Prediction.** The drug-drug interaction prediction task is regarded as a multirelational link prediction problem. Inspired by the loss design in [Zitnik et al., 2018], we train the parameters with the following cross-entropy loss function

\[ p_{ij}^r = \sigma((W_r x_{G_i})^T (W_r x_{G_j})) \]  

(9)

\[ \mathcal{L}_{ij}^r = -\log(p_{ij}^r) - \mathbb{E}_{m \sim P^r_{im}} \log(1 - p_{ij}^m) \]  

(10)

\[ \mathcal{L}_{DDI} = \sum_{(G_i, r, G_j) \in \mathcal{G}_{DDI}} \mathcal{L}_{ij}^r \]  

(11)

where \( W_r \) is the side-effect-specific weight for linear transformation of \( x_{G_i} \). Given observed triplet \((G_i, r, G_j)\), the negative sample is chosen by replacing \( G_j \) with randomly selected graph \( G_m \) according to sampling distribution \( P^r_m \) [Mikolov et al., 2013].

5 Experiment

In this section, we introduce the extensive experiment results that demonstrate the effectiveness and robustness of GoGNN.

5.1 Dataset

To test the performance of our model on chemical-chemical interaction and drug-drug interaction prediction tasks, following datasets are chosen for the experiments:

CCI. The CCI dataset\(^2\) assigns a score from 0 to 999 to describe the interaction probability where a higher score indicates higher interaction probability. According to threshold score, we get two datasets with chemical interaction probability score over 900 and 950: CCI900 and CCI950. CCI900 has 14343 chemicals and 110078 chemical interaction edges, and CCI950 has 7606 chemicals and 34412 chemical interaction edges.

DDI. For the drug-drug interaction prediction problem, DDI dataset\(^1\) and the side effect dataset SET\(^4\) [Zitnik et al., 2018] are used. The DDI dataset is proposed by DeepDDI [Ryu et al., 2018] which contains 86 types of side effects, 1704 drugs and 191400 drug interaction edges. SE dataset is the integration of SIDER (Side Effect Resource), OFFSIDES and TWOSIDES database. To familiarize the comparison, we use the preprocessed data used by Decagon [Zitnik et al., 2018]. Therefore, the SE dataset contains 645 drugs, 964 types of side effects and 4651131 drug-drug interaction edges. A vector representation \( x_e \in \mathbb{R}^{128} \) is assigned to each side effect type produced by pre-trained BERT model [Devlin et al., 2018].

The molecules are transformed from the SMILE strings [Weininger et al., 1989] into graphs by the opensource rdkit [Landrum, 2013]. An initial feature vector \( x_o \in \mathbb{R}^{32} \) is assigned for every atom. The edges in molecule graphs are weighted by the type of the bonds.

5.2 Baselines

The proposed GoGNN is compared with the following state-of-the-art models:

DeepCCI [Kwon and Yoon, 2017] is the CNN based model for predicting the interactions between the chemicals.

DeepDDI [Ryu et al., 2018] is the model designs a feature called structural similarity profile(SSP) combined with traditional MLP for DDI prediction.

Decagon [Zitnik et al., 2018] is a GCN model on the drug and protein interaction graphs to predict the polypharmacy side effects caused by drug combinations.

MR-GNN [Xu et al., 2019] is an end-to-end graph neural network with multi-resolution architecture that produces interaction between pairs of chemical graphs.

MLRDA [Chu et al., 2019] is the multitask, semi-supervised model for DDI prediction.

SEAL [Li et al., 2019a] is the neural network on hierarchical graphs for graph classification.

We used the public code of the baselines and keep the settings of models the same as mentioned in the original papers. We reimplemented SEAL for CCI and DDI prediction.

Ablation Study

To investigate how the graph of graphs architecture and dual-attention mechanism improve the performance of the proposed model, we conduct the ablation study on the following variants of GoGNN:

GoGNN-M is the variant which only learns the representations for the molecule-level graphs without the graph convolution on the interaction graph. An MLP layer is applied with the input of molecule-level graph representations for the graph interaction prediction task.

GoGNN-I only conducts graph convolution operation on the chemical interaction graphs. The initial molecule representations are the sum pooling of the atom representations within the molecule.

GoGNN-noPool replaces the self-attention pooling on the molecule graph by the concatenation of conventional mean pooling and sum pooling.

GoGNN-noAttn replaces the attention-based neural network on the interaction graph by a conventional GCN.

5.3 CCI Prediction Results

Settings. Following the previous study, we divide the CCI datasets into training and testing set with ratio 9:1, and randomly choose 10% data for validation. The dimensions of molecule graph hidden feature, and the output molecule graph representation are set to 384, 256, respectively. We set the learning rate to 0.01 and the pooling ratio to 0.5. To evaluate the performance, we choose area under the ROC curve (AUC) and average precision score (AP) as metrics.

Results. As shown in Table 1, GoGNN outperforms all the other state-of-the-art baseline methods on the CCI prediction task. The improvement indicates that, compared with the methods that only train the parameters with pair-wise or individual chemical inputs, GoGNN can preserve more useful information on different scales by the feature extraction and aggregation through the graph of graphs. The dual-attention mechanism also helps the model to learn higher quality graph representations by identifying and preserving the importance of molecular substructures and chemical interactions.
GoGNN reaches the best performance with representation dimension 128. Figure 3b indicates that the salient point for the hidden feature size is 384. As for the learning rate and pooling ratio, the best point appears at 0.001, pooling ratio \( \tau = 0.5 \). We choose \( \text{AUC} \) and \( \text{average precision}(\text{AP}) \) for evaluation.

### 5.4 DDI Prediction Results

**Settings.** To familiarize the comparison, we divide the DDI dataset for training, testing, validation with ratio 6:2:2, and divide the SE dataset with ratio 8:1:1. The dimensions of molecule graph hidden feature, and the output molecule graph representation are set to 384, 256, respectively. We set the learning rate to 0.001, pooling ratio \( \tau = 0.5 \). We choose \( \text{AUC} \) and \( \text{average precision}(\text{AP}) \) for evaluation.

**Results.** The experiment results for DDI prediction are listed in Table 2. The results show that compared with the baseline methods, GoGNN improves the performance with a significantly large margin. GoGNN improves the AUC and AP by 1.18% and 1.19% respectively on DDI dataset, and 6.65% and 11.42% respectively on SE dataset. The improvement is attributed to the abundant information brought by the graph of graphs architecture and edge-filtered aggregation.

### 5.5 Ablation Experiments

The ablation experiment results on both tasks are shown in Table 1 and Table 2. The results prove that the graph of graphs architecture, attention-based pooling, attention-based and edge-filtered aggregation are all effective for the side-effect prediction task. Among all the variants, GoGNN-M and GoGNN-I have the most significant performance gaps between GoGNN, which indicates that the view of graph of graphs contributes the most to helping the model to capture more structural information that improves the prediction accuracy.

### 5.6 Parameter Sensitivity Analysis

In this experiment, we test the impact of the hyper-parameters of GoGNN.

**Settings.** We conduct the parameter sensitivity experiment on the CCI950 dataset by changing the tested hyper-parameter while keeping other settings the same as mentioned in Section 5.3. We test the following hyper-parameters: the dimensions of the output representation and hidden feature, learning rate and pooling ratio.

**Results.** As shown in Figure 3, overall, the impact of hyperparameter variation is insignificant. Figure 3a shows that GoGNN reaches the best performance with representation dimension 128. Figure 3b indicates that the salient point for the hidden feature size is 384. As for the learning rate and pooling ratio, the best point appears at \( 10 \times 10^{-3} \) and 0.5, respectively.

### 6 Conclusion

In this paper, we focus on structured entity interaction prediction. This prediction demands the model to capture the information of the structure of entities and the interactions between entities. However, the previous works represent the entities with insufficient information. To address this limitation, we propose a novel model GoGNN which leverages the dual-attention mechanism in the view of graph of graphs to capture the information from both entity graphs and entity interaction graph hierarchically. The experiments on real-life datasets demonstrate that our model could improve the performance on the chemical-chemical interaction prediction and drug-drug interaction prediction tasks. GoGNN can be naturally extended to the applications on other graph of graphs such as financial networks, electrical networks, etc. We leave the extension for future work.

---

### Table 1: Result of chemical-chemical interaction prediction task.

<table>
<thead>
<tr>
<th></th>
<th>CCI900</th>
<th></th>
<th>CCI950</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>AP</td>
<td>AUC</td>
<td>AP</td>
</tr>
<tr>
<td>DeepCCI</td>
<td>0.925</td>
<td>0.918</td>
<td>0.957</td>
<td>0.957</td>
</tr>
<tr>
<td>DeepDDI</td>
<td>0.891</td>
<td>0.886</td>
<td>0.916</td>
<td>0.915</td>
</tr>
<tr>
<td>MR-GNN</td>
<td>0.927</td>
<td>0.921</td>
<td>0.934</td>
<td>0.924</td>
</tr>
<tr>
<td>MLRDA</td>
<td>0.922</td>
<td>0.907</td>
<td>0.959</td>
<td>0.948</td>
</tr>
<tr>
<td>SEAL</td>
<td>0.894</td>
<td>0.886</td>
<td>0.941</td>
<td>0.937</td>
</tr>
<tr>
<td>GoGNN</td>
<td>0.937</td>
<td>0.932</td>
<td>0.963</td>
<td>0.962</td>
</tr>
<tr>
<td>GoGNN-M</td>
<td>0.914</td>
<td>0.909</td>
<td>0.938</td>
<td>0.931</td>
</tr>
<tr>
<td>GoGNN-I</td>
<td>0.921</td>
<td>0.898</td>
<td>0.929</td>
<td>0.912</td>
</tr>
<tr>
<td>GoGNN-noPool</td>
<td>0.931</td>
<td>0.930</td>
<td>0.958</td>
<td>0.954</td>
</tr>
<tr>
<td>GoGNN-noAttn</td>
<td>0.909</td>
<td>0.905</td>
<td>0.956</td>
<td>0.948</td>
</tr>
</tbody>
</table>

* indicates that the result is the output of the baselines after two weeks’ training.

\* DDI dataset has no protein data which is required by Decagon

---

### Table 2: Result of drug-drug interaction prediction task.

<table>
<thead>
<tr>
<th></th>
<th>DDI</th>
<th></th>
<th>SE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>AP</td>
<td>AUC</td>
<td>AP</td>
</tr>
<tr>
<td>DeepCCI</td>
<td>0.862</td>
<td>0.856</td>
<td>0.819</td>
<td>0.806</td>
</tr>
<tr>
<td>DeepDDI</td>
<td>0.915</td>
<td>0.912</td>
<td>0.827</td>
<td>0.809</td>
</tr>
<tr>
<td>MR-GNN</td>
<td>0.932</td>
<td>0.922</td>
<td>0.769*</td>
<td>0.752*</td>
</tr>
<tr>
<td>MLRDA</td>
<td>0.931</td>
<td>0.926</td>
<td>0.847*</td>
<td>0.825*</td>
</tr>
<tr>
<td>Decagon</td>
<td>-</td>
<td>-</td>
<td>0.872</td>
<td>0.832</td>
</tr>
<tr>
<td>SEAL</td>
<td>0.925</td>
<td>0.921</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>GoGNN</td>
<td>0.943</td>
<td>0.933</td>
<td>0.930</td>
<td>0.927</td>
</tr>
<tr>
<td>GoGNN-M</td>
<td>0.905</td>
<td>0.902</td>
<td>0.862</td>
<td>0.817</td>
</tr>
<tr>
<td>GoGNN-I</td>
<td>0.922</td>
<td>0.917</td>
<td>0.860</td>
<td>0.834</td>
</tr>
<tr>
<td>GoGNN-noPool</td>
<td>0.900</td>
<td>0.891</td>
<td>0.912</td>
<td>0.909</td>
</tr>
<tr>
<td>GoGNN-noAttn</td>
<td>0.925</td>
<td>0.921</td>
<td>0.897</td>
<td>0.883</td>
</tr>
</tbody>
</table>

---

Figure 3: Parameter sensitivity experiment results...
References


