Noname manuscript No.
(will be inserted by the editor)

Powerful Graph of Graphs Neural Network for Structured Entity Analysis

**Hanchen Wang · Defu Lian · Wanqi Liu ·
Dong Wen · Chen Chen · Xiaoyang Wang**

Received: date / Accepted: date

Abstract Structured entities analysis is the basis of the modern science, such as chemical science, biological science, environmental science and medical science. Recently, a huge amount of computational models have been proposed to analyze structured entities such as chemical molecules and proteins. However, the problem becomes complex when local structural entity graphs and a global entity interaction graph are both involved. The unique graph of graphs structure cannot be properly exploited by most existing works for structural entity analysis. Some works that build neural networks on the graph of graphs cannot preserve the local graph structure effectively, hence, reducing the expressive power of the model. In this paper, we propose a Powerful Graph Of graphs neural Network, namely PGON, which has 3-Weisfeiler-Lehman expressive power and captures the attributes and structural information from both structured entity graphs and entity interaction graph hierarchically. Extensive experiments are conducted on real-world datasets, which show that PGON

Hanchen Wang
University of Technology Sydney
E-mail: hanchenw.au@gmail.com

Defu Lian
University of Science and Technology of China
E-mail: liandefu@ustc.edu.cn

Wanqi Liu
University of Technology Sydney
E-mail: wanqi.liu@student.uts.edu.au

Dong Wen
University of Technology Sydney
E-mail: dong.wen@uts.edu.au

Chen Chen
Zhejiang Gongshang University
E-mail: chenc@zjgsu.edu.cn

Xiaoyang Wang
Zhejiang Gongshang University
E-mail: xiaoyangw@zjgsu.edu.cn

outperforms other state-of-the-art methods on both graph classification and graph interaction prediction tasks.

Keywords Graph Neural Network · Graph of Graphs · Weisfeiler-Lehman Test · Graph Classification

1 Introduction

Structured entities such as chemical molecules and proteins are frequently used in many fields such as biology, chemistry, material science, medical science and environmental science. The analysis of structured entities is the basis of our life. For instance, new material is developed based on the analysis of existing chemicals and new drugs are discovered based on the interaction between drugs and proteins and the properties of other drugs.

One immediate way to analyze structured entities is to conduct experiments to test the properties of such entities or the interactions between entities in laboratories or clinics. This is the most reliable method for structured entity analysis. Nevertheless, because of the large number of structured entities, it is impracticable with regard to both resources and time to investigate all entities.

Recently, structured entities have been modeled as graphs and analyzed by many computational algorithms, such as graph classification, link prediction, clustering, etc. For example, graph classification results could help to determine the function of proteins and the toxicity of chemical compounds. The link prediction between the graphs could be used to understand the interactions between chemical molecules, drugs or proteins, which are vital for drug discovery, side effect prediction and pathology analysis [29]. Therefore, DeepCCI [18] and DeepDDI [35] are proposed to predict the interactions between chemicals and drugs respectively.

Thanks to the advancement of graph neural networks, a variety of techniques have been proposed to analyze structured entities effectively and efficiently by utilizing graph neural networks (GNNs). Therefore, instead of conducting time-consuming and labor intensive laboratory experiments, the GNN-based models provide an efficient and effective solution for the analysis of biological and chemical data. With the development of GNNs, structural and feature information are well exploited in such analysis.

Decagon [49] exploits the drug-drug, drug-protein and protein-protein network to predict drug side effects; MR-GNN [43] utilizes the multi-resolution graph neural network to model the detailed structure of the chemicals; and GCPN [45] proposed a graph convolutional policy network to generate molecular graphs.

Though the structural information of a graph acts as a pivotal part of GNN-based models, it has not been extensively studied yet. Most GNN-based models only exploit graph structural information and ignore the interaction information between the graphs. For example, in the protein interaction network, conventional GNN-based models learn representations only from the structural information of the proteins or from the interaction relations between the proteins. In Fig. 1, there is a graph containing the interactions between the chemical compounds. The chemical molecules are modeled as *local* graphs. The atoms in the molecule are modeled as nodes in

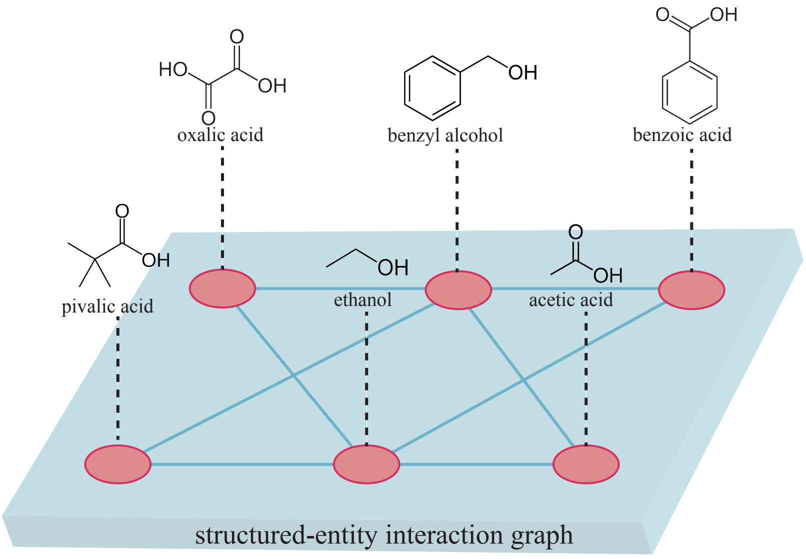


Fig. 1: Interaction graph of molecule graphs.

the graph, while the bonds between the atoms are represented as the edges. On the other hand, we have a *global* graph which contains the interactions (edges) among the structured entities (nodes). Only a few works combine these two kinds of structures together and produce the more informative representations for down-stream graph analysis tasks. SEAL-CI [22] applies a hierarchical graph neural network to capture the graph interaction information. DGCN [15] learns the compound representations from both the compound graphs and the inter-compound network in an end-to-end manner. The work in [40] introduces GoGNN, the graph of graphs neural network, to solve graph interaction prediction tasks. However, these methods cannot powerfully and distinguishably represent graph structural information together with graph features. Further, they focus on only one task, i.e., link prediction or graph classification, which limits their generalizability.

Motivated by the shortcomings of the current models, we propose our **P**owerful **G**raph **O**f graphs neural **N**etwork (PGON) to capture graph interaction and graph structural information for multiple graph analysis tasks. PGON uses the invariant and equivariant functions to build the graph neural network following the 2-folklore Weisfeiler-Lehman (FWL) isomorphism test algorithm such that PGON is able to powerfully preserve the structural information for each molecular or protein graph. PGON also follows the graph of graphs framework. In that way, PGON could consider both levels of the graphs to extract broader information, which improves the performance of PGON on graph analysis tasks. The contributions of our proposed model are summarized as follows:

- We exploit the equivariant and invariant mapping functions to build the graph neural network following the 2-FWL test. Therefore, our model is able to represent the graph structure as powerfully as a 3-Weisfeiler-Lehman test algorithm.
- We study both graph classification and graph interaction prediction tasks from a graph of graphs perspective. Our model could exploit more topological information and the interactions between local graphs to enhance performance.
- We evaluate our proposed PGON using both graph classification and graph interaction prediction tasks on real-world datasets. PGON has superior performance compared to other baseline methods.

2 Related Work

In this section, we introduce the related works including graph neural networks, graph of graphs and structured entity analysis models.

2.1 Graph Neural Network

Node-level applications. There are two main kinds of node-level applications of graph neural networks: node classification and link prediction [17,38,46,13,27,25,24,14,39,37]. In order to preserve the relations among the nodes in the graph, structural and attribute information of the graph and represent the graph by low-dimension vectors, the existing node embedding methods mainly exploits the conventional techniques such as skip-gram, autoencoders and neighbor aggregation-based graph neural networks.

Graph-level applications. Recently, several research papers proposed graph neural network-based models for graph-level applications such as graph classification [47,21,48,32,9] and graph matching[23,10]. These models produce the graph representations only considering the individual or pair-wise graph relations, and ignoring the interactions between the graphs. Therefore, they cannot aggregate all the information from the neighboring graphs simultaneously, which limits their performance on these tasks.

2.2 Graph of Graphs

We first develop the concept of “Graph of Graphs” (a.k.a., “Network of Networks”). In most real-world systems, an individual network is one component within a much larger complex multi-level network. These networks are network of networks (graph of graphs) when the graph theory paradigm is applied. [4] introduces the theoretical research development [8], applications [33] and phenomenological model [34] on the graph of graphs.

These works enable us to understand and model the inter-dependent critical infrastructures. The work in [22] proposed SEAL-CL which solves the graph classification task by learning a graph neural network in a hierarchical graph perspective.

DGCN [15] is a dual graph convolutional network that learns compound representations from both the compound graphs and the inter-compound network in an end-to-end manner. GoGNN [40] proposed the graph of graphs neural network to predict the interactions between structured entities.

2.3 Structured Entity Analysis

Many modern science are based on the analysis of structured entities, such as chemical science, biological science, medical science, and environmental science. Therefore, we need to capture the property of the entities and structured entity interactions. Over the last few years, a great amount of structured entity analysis techniques have been proposed in some specific applications.

GCPN [45] proposed the graph convolutional policy network to generate molecular graphs. GRAN [26] exploits the graph recurrent attention networks for the graph generation. CONDGEN [44] generates the conditional graph structure by graph variational generative adversarial network. As for the graph interaction prediction task, DeepCCI and DeepDDI [18,35] utilize traditional convolutional neural network and principal component analysis on the medical and biological data. There are some GNN-based models. For instance, Decagon [49] performs the graph convolutional network on the drug-protein interaction graph; MLRDA [3] utilizes a graph autoencoder model with a novel loss function for drug-drug interaction prediction task; MR-GNN [43] extracts local features of molecule graphs using dual graph-state LSTMs.

In this paper, two representative applications are studied: structured entity classification and structured entity interaction prediction.

3 Preliminary

In this part, we introduce the important definitions and the notations used in this paper.

3.1 Input Graphs

In this paper, we not only focus on graph structures but also consider the interactions between the graphs. Therefore, the input graphs are regarded as graph-of-graphs. There are two hierarchical structures of graph-of-graphs: local graphs which illustrate the local information of the objects involved in the interaction network, and interaction graph describes the interaction relationships between the local instances.
Local Graphs. Local graphs are graph instances in this paper. More specifically, local graphs are molecule graphs which are modelled as heterogeneous graphs with multiple types of nodes and edges or the protein graphs containing amino acids and their connection relationships. We formally denote the local graph as $G_L(V, E)$ where $V = \{v_i\}$ is the set of vertexes (atoms or amino acids) in the graphs and $E = \{e_{ij}\}$ is the set of edges representing the bonds between atoms or the connection between amino acids v_i and v_j. As the input, a vector $\boldsymbol{x}_a$ is used to encode the vertex. In the

molecule graphs, a weight is assigned to each edge which is determined by the kind of the bond. For instance, in the acetylene molecule, the bond between the carbon atoms is a triple bond. As a result, we set the weight of the edge e_{CC} in the acetylene molecule graph between the carbon atoms to 3.

Interaction Graphs. We model the interactions between the local graphs as a graph G_I. Thus, we have the graph of graphs. The interaction graph is denoted by $G_I = \{N, E_I\}$, where $N = \{G_L\}$ is the node set of G_I that contains the local graphs, and E_I denotes the interaction edges between the local graphs.

Example 1 (Graph of Graphs.) There are various graph of graphs in real life. We list one representative example here. The software interaction networks for malware detection is one typical graph of graphs. The local graphs in the software interaction networks are the function call graphs, which represent calling relationships between subroutines in a computer programs. The interaction graph illustrates the interaction between the computer programs involved in the network.

3.2 Problem Definition

In this paper, we focus on the graph classification and graph interaction prediction task.

Graph Classification: a molecule graph g_L is called a *labeled graph* if it is labeled by a class vector $\boldsymbol{y}_L \in \{0,1\}^c$, where c denotes the number of classes. Otherwise, a local graph is *unlabeled*. Therefore, we have two subsets of graphs: unlabeled graphs G_U and labeled graphs G_{Label}. The objective of **graph classification** is to determine the class labels of the unlabeled graphs in G_U with given class labels in the labeled graphs G_{Label} and the graph of graphs topological structural information.

Graph Interaction Prediction: Given the local graph $g = (N, E)$, and graph interaction graph $G = (g, I)$, where I indicates the interactions between the local graphs g, the objective of **graph interaction prediction** is to predict the existence of unseen interactions I_u between the graphs.

3.3 Theoretical Definitions

Definition 1 Equivariant Function. With given matrix $\boldsymbol{X}$ and any permutation matrix $\boldsymbol{P}$, a function f is an equivariant function if $f(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{P}) = \boldsymbol{P}^T f(\boldsymbol{X})\boldsymbol{P}$.

Definition 2 Invariant Function. With given matrix $\boldsymbol{X}$ and any permutation matrix $\boldsymbol{P}$, a function f is an invariant function if $f(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{P}) = f(\boldsymbol{X})$.

3.4 The Weisfeiler-Lehman graph isomorphism test.

The Weisfeiler-Lehman (WL) graph isomorphism test is a family of algorithms used to test the graph isomorphism. Given graph $G(V, E, d)$ where $|V| = n$, and $d : V \to$

Σ maps each vertex in V to the color set Σ. Two graphs are isomorphic if there exists a bijection $\phi : V \rightarrow V'$ that preserves the edge and color attached to the vertices.

There are two families of WL-test algorithms: k-WL and k-Folklore WL (FWL) algorithms. They both produce the canonical form for each graph G and construct the k-tuples of vertices with the corresponding color set, that is mapping $c : V^k \rightarrow \Sigma$. $\boldsymbol{C}$ is the tensor representing the coloring of k-tuples. For each k-tuple $v_i = (v_{i_1}, ..., v_{i_k}) \in V^k$, the corresponding color set is denoted as $\boldsymbol{C}_i \in \Sigma$, $i \in \binom{n}{k}$.

The k-WL and k-FWL test algorithms refine the coloring set $\boldsymbol{C}$ in every iteration until the split groups of k-tuples no longer change. In each step, the k-WL and k-FWL algorithms aggregate the color labels from the neighboring k-tuples to update the coloring $\boldsymbol{C}$. The aggregation steps for the algorithms to refine the coloring set of each k-tuple are shown in the following equations:

$$N_j(i) = \{(i_1, ..., i_{j-1}, i', i_{j+1}, ..., i_k) | i' \in [n]\} \tag{1}$$

$$N_j^F(i) = \{(j, i_2, ..., i_k), (i_1, j, ..., i_k), ..., (i_1, ..., i_{k-1}, j)\} \tag{2}$$

$N_j(i), j \in [k]$ is a set of n k-tuples denoting the j-th neighborhood of the selected tuple i by the k-WL algorithm. $N_j^F(i), j \in [n]$ denotes the j-th neighborhood used by k-FWL algorithm. Then both k-WL and k-FWL algorithms aggregate the coloring set using the bijective mapping from the collection of all possible tuples selected by Equations 1 and 2.

Previous works [2,11,12,28] show that 1-WL and 2-WL tests have equivalent discrimination power. When $k \geq 2$, the $k+1$-WL test is more powerful than the k-WL test in detecting non-isomorphic graphs.

4 Model

In this section, we introduce details of the powerful graph of graphs neural network (PGON). The framework, graph of graphs architecture and the training objectives are discussed in this section.

4.1 Framework

PGON is a graph neural network model that takes the graph of graphs structure and feature information as input, and then produces the predicted class labels or predicts the interaction probabilities between the graphs in an end-to-end manner. The framework of PGON is shown in Figure 2. The model is formed by two graph neural networks. A local graph neural network whose input is the atom or amino acid features. An interaction graph neural network which eventually produces a representation of graph for down-stream tasks, i.e., graph classification and graph interaction prediction. Local graph neural network learns the hidden features which serve as the initial input for the interaction graph neural network. The hidden feature is representative since we exploit the GNN that has the same expressive power as 3-WL tests. The

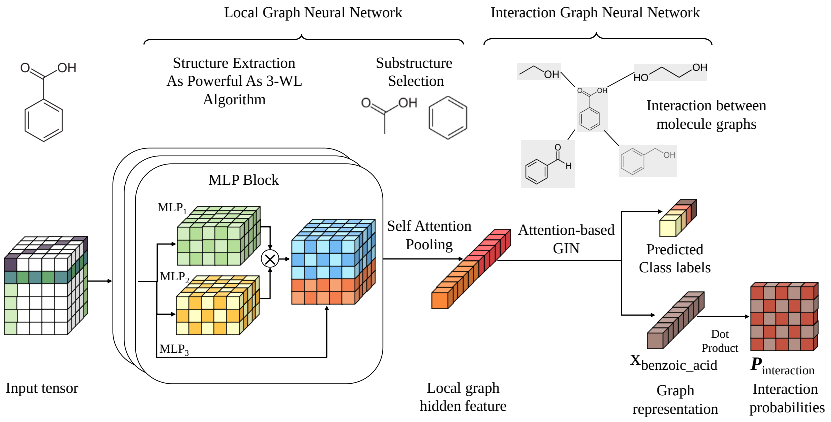

Local Graph Neural Network
Interaction Graph Neural Network
Structure Extraction As Powerful As 3-WL Algorithm
Substructure Selection
Interaction between molecule graphs
MLP Block
MLP1
MLP2
MLP3
Self Attention Pooling
Attention-based GIN
Predicted Class labels
Dot Product
Input tensor
Local graph hidden feature
X benzoic_acid
Graph representation
P interaction
Interaction probabilities

Fig. 2: Framework of Powerful Graph of Graphs Neural Network.

interaction-level graph neural network promotes the ability of local graph neural network to find the key substructure through training process by the feature aggregation mechanism.

4.2 Local Graph Neural Network

The input local graphs are chemical or protein graphs. In these kinds of graphs, the structure of the graph is very important for the analysis task. For example, the reactions between the organic chemical molecules are mainly decided by some specific subgraphs: functional groups. Reactions between organic acid and ethanol are intrinsically controlled by two functional groups, *i.e.,* the carboxy group (-COOH) and hydroxy group (-OH). As for the proteins, the structures of proteins, especially the sequences and structures of amino-acids, are also responsible for the characteristic properties of the proteins. Though currently we cannot model the 3D structure of proteins using graph neural networks, the discriminative GNN model could preserve the plain structures of the proteins which are the most important for protein properties.

Based on the above motivations, we design the local graph neural network which is discriminative enough to distinguish graph structures. Therefore, we use the graph neural network proposed by Maron et al. [28]. The GNN model has 3-WL discrimination power and low computation complexity. The model is formed by the invariant layer and several MLP blocks:

$$F = m \circ h \circ M_k \circ M_{k-1} \circ ... \circ M_1, \tag{3}$$

where m is a transformation layer such as MLP, h is an invariant layer and $M_1, ..., M_k$ are the MLP-based blocks described below. Each block has three MLPs, two of them (MLP_1 and MLP_2) map the tensor dimensions $\mathbb{R}^{n\times n\times d_0} \rightarrow \mathbb{R}^{n\times n\times d_1}$, and the other transforms the tensor dimensions $\mathbb{R}^{n\times n\times d_0} \rightarrow \mathbb{R}^{n\times n\times d_2}$. The detail of the input tensor is introduced in Section 5. The tensors produced by the MLP_1 and MLP_2

are in the same dimensions. These two tensors are multiplied to produce the new feature tensor. The computation process for each MLP block is shown in the following equations:

$$\boldsymbol{X}_i = MLP_i(\boldsymbol{X}), \{i = 1, 2, 3\} \tag{4}$$

$$\boldsymbol{X}_4 = \boldsymbol{X}_1 \otimes \boldsymbol{X}_2 \tag{5}$$

$$\boldsymbol{X}_{out} = \boldsymbol{X}_3 \oplus \boldsymbol{X}_4, \tag{6}$$

where $\otimes$ indicates the matrix multiplication between the corresponding matrices: $\boldsymbol{X}_{1(:,:,j)}$ and $\boldsymbol{X}_{2(:,:,j)}$ and $\oplus$ is the concatenation of the tensors. $\boldsymbol{X}_{out}$ is the output feature tensor for every graph after the MLP block. Different from MR-GNN [43] which uses dual graph-state LSTMs, PGON exploits pooling-based graph model to learn the representation that preserves the substructure information of each graph. Therefore, PGON enjoys lower time and space complexity compared to MR-GNN. In the model of PGON shown in Figure 2, the most representative substructures are selected in the self-attention graph pooling layer by learning the self-attention score $\boldsymbol{s} \in \mathbb{R}^{n \times 1}$ for the local graph G_L with n vertices.

$$\boldsymbol{s} = \sigma(\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{X}_{feat} \boldsymbol{W}_{att}) \tag{7}$$

where $\boldsymbol{W}_{att} \in \mathbb{R}^{d \times 1}$ denotes the weight matrix in the self-attention pooling layer, and $\boldsymbol{X}_{feat}$ denotes the feature vectors of the graph on the diagonal of the output tensor $\boldsymbol{X}_{out i,i,:}$. After the calculation of attention score in the graph pooling layer, the most representative substructure is selected by finding the top-$\lceil \gamma n \rceil$ vertices with the highest attention scores. A hyperparameter $\gamma \in (0, 1]$ to is used as the pooling ratio to determine the number of vertices $\lceil \gamma n \rceil$ that are selected.

$$\begin{gathered} idx = top(\boldsymbol{s}, \lceil \gamma n \rceil), \boldsymbol{s}_{mask} = \boldsymbol{s}_{idx} \\ \boldsymbol{X}_{sel} = \boldsymbol{X} \odot \boldsymbol{s}_{mask} \end{gathered} \tag{8}$$

where top denotes the function selecting the vertices that have top $\lceil \gamma n \rceil$ attention scores as in [20]; $\boldsymbol{s}_{mask}$ denotes a mask vector decided by the attention score; $\odot$ is the column-wise product for masking; and $\boldsymbol{X}_{sel}$ denotes the feature vectors of the chosen nodes within a local graph. Afterward, the readout layer, which contains sum and mean pooling, is performed on the representations of the chosen nodes $\boldsymbol{X}_{sel}$ to learn the graph-level hidden feature vector $\boldsymbol{x}_{sel}$ for the diagonal elements from the tensor $\boldsymbol{X}_{out}$. As for the off-diagonal elements in the tensor $\boldsymbol{X}_{out}$, max pooling is applied to produce the hidden feature $\boldsymbol{x}_{off} \in \mathbb{R}^{d_{off}}$ for the local graph. The hidden feature $\boldsymbol{x}_{feat}$ is the concatenation of $\boldsymbol{x}_{sel}$ and $\boldsymbol{x}_{off}$. Hence, PGON can identify the substructures which contribute significantly in graph interactions, and represent the local graph using these substructures.

4.3 Interaction Graph Neural Network

In the graph analysis tasks, the interactions between the graphs are always ignored. However, it is crucial for graph analysis to obtain the information of the interaction graph because the information enables the model to capture high-order interactivity relationships. Therefore, PGON has a better ability to preserve the characteristic molecule substructures synergistically which benefits both graph classification and graph interaction prediction tasks.

We perform the GNN on the interaction graph based on the following observations: firstly, the type of the involved molecules in an interaction determines the type of interaction. For example, Diels-Alder reaction is a kind of organic chemical reaction between a conjugated diene and a substituted alkene. Specifically, the Diels-Alder reaction is between the two double bonds in the conjugated diene and the double bond in the substituted alkene. Therefore, we use the graph neural network to aggregate the neighbor information, and eventually conclude the types of molecules involved in the interaction graph. Secondly, in order to model the frequency and significance of the interactions, the significance score is assigned to each interaction of the graph. For instance, reducibility and acidity are main properties of vitamin C. Because of the properties, vitamin C can never be taken with procarbazine and sucralfate due to their oxidization and alkalinity respectively. In a much rarer situation, VC reduces the therapeutic effect of inosine due to their multiplex reactions. As a result, an attention-based graph isomorphism network [42] is applied to preserve the significance and importance of the graph interactions, reduce the influence of biased observations of the interaction graph and maintain the representation power of the model. Specifically, the edge-aggregation GNN is exploited for the DDI graph with edge attributes.

Attention-based GIN. We add the attention coefficients on the original GIN [42] to assign importance values to graph interactions. With the output local graph hidden feature vector $\boldsymbol{x}_{G_L}$ and interaction graph $G_I = \{N, E_I\}$ as the input, the attention-based graph isomorphism network perform the neighbor aggregation as follows:

$$\boldsymbol{x}^k_{G_i} = MLP^k((1+\epsilon^k)\boldsymbol{x}^{k-1}_{G_i} + \sum_{G_j \in \mathcal{N}(G_j)} a^k_{ij}\boldsymbol{x}^{k-1}_{G_j}). \tag{9}$$

In the equation, ϵ is a learnable parameter to adjust the importance of the graph representation, and a^k_{ij} is the attention coefficient between two graphs G_i and G_j. The attention coefficient is calculated by the following equation:

$$a_{ij} = \frac{exp(\text{LeakeyRelu}(\boldsymbol{a}[\boldsymbol{W}\boldsymbol{x}_{G_i}] \parallel [\boldsymbol{W}\boldsymbol{x}_{G_j}]))}{\Sigma_{n\in\eta_{G_i}} exp(\text{LeakeyRelu}(\boldsymbol{a}[\boldsymbol{W}\boldsymbol{x}_{G_i}] \parallel [\boldsymbol{W}\boldsymbol{x}_{G_n}]))} \tag{10}$$

where $\boldsymbol{a}$ denotes a learnable vector indicating the attention weights and $\parallel$ denotes the concatenation operation.

Edge Aggregation Network. In the drug-drug interaction graph, each edge has an attribute vector $\boldsymbol{e}^r_{ij}$ which is determined by the side effect type r of the drug combination (G_i, G_j). To capture the edge attributes [36], an edge aggregation GNN is proposed. The network aggregates the neighbor information together with the edge attribute:

$$\boldsymbol{x}^{l+1}_{G_i} = \sigma(\boldsymbol{W}^l\boldsymbol{x}^l_{G_i} + \sum_r(\sum_{G_j\in\eta^r_{G_i}} \boldsymbol{x}^l_{G_j} \cdot h_{\boldsymbol{W}_{\mathbf{e}}}(\boldsymbol{e}^r_{ij}))) \tag{11}$$

where $h_{\boldsymbol{W}_e}$ denotes the multi-layer perceptron with matrix $\boldsymbol{W}_e$ which linearly transforms the edge feature representation $\boldsymbol{e}^r_{ij} \in \mathbb{R}^{h\times 1}$ into a real number $\tau^r_{ij} \in \mathbb{R}$. PGON shares the parameters for all types of side effects which is different from Decagon [49] whose parameters are set for each side effect specifically. Therefore, PGON is the model that has better generalization and robustness.

4.4 PGON Model Training

The parameters in the model are optimized by task-specific loss functions.

Graph Classification. We apply MLPs with activations to transfer the learned graph representations $\boldsymbol{x}_{G_i}$ into the labels:

$$y_{G_i} = \sigma(W^l \cdot \sigma(\cdots W^1 \cdot \boldsymbol{x}_{G_i})), \tag{12}$$

where σ denotes the activation function such as softmax, and W is the weight matrix in MLP. With the obtained class labels, the cross-entropy is proven to be a good choice to measure the loss of the predicted labels:

$$\mathcal{L}_{class} = \sum_{G_i \in G_{labeled}} \mathcal{J}(\hat{y}_{G_i}, y_{G_i}), \tag{13}$$

where $\mathcal{J}$ is the cross-entropy loss and $\hat{y}_{G_i}$ is the ground truth labels of the labeled graphs. The classification model is trained by minimizing the above loss function.

Chemical Interaction Prediction. The CCI prediction is modeled as a link prediction task. We simulate the probability of linkage between two graphs by the dot product of two graph embedding vectors.

$$p_{ij} = \sigma(\boldsymbol{x}^T_{G_i} \cdot \boldsymbol{x}_{G_j}) \tag{14}$$

where σ denote the activation function such as *tanh* which constraints $p_{ij} \in (0, 1)$. It is natural that the existing edges should have higher significance scores compared to the random non-edges. For this purpose, negative sampling is utilized. Given a positive edge pair (G_i, G_j), a random negative edge (G_i, G_m) is sampled by selecting a local graph G_m randomly. The cross-entropy loss function is used for the optimization of the model

$$\mathcal{L}_{CCI} = \sum_{(G_i,G_j)\in G_{CCI}} -log(p_{ij}) - \mathbb{E}_{m\sim P_j} log(1 - p_{im}) \tag{15}$$

Drug Interaction Prediction. Because there are many kinds of adverse effects in drug-drug interaction graph. The DDI prediction task is modeled as a multirelational link prediction problem. The following cross-entropy loss function is utilized to optimize the parameters in this task

$$p^r_{ij} = \sigma((\boldsymbol{W}_r \boldsymbol{x}_{G_i})^T \cdot (\boldsymbol{W}_r \boldsymbol{x}_{G_j})) \tag{16}$$

$$\mathcal{L}^r_{ij} = -log(p^r_{ij}) - \mathbb{E}_{m\sim P^r_j} log(1 - p^r_{im}) \tag{17}$$

$$\mathcal{L}_{DDI} = \sum_{(G_i,r,G_j)\in G_{DDI}} \mathcal{L}_{ij}^{r} \tag{18}$$

where $\boldsymbol{W}_r$ is a weight matrix for the linear transformation of $\boldsymbol{x}_{G_i}$ w.r.t. the side effect type r. For an existing triplet (G_i, r, G_j) denoting two drug graphs G_i and G_j which cause a side effect r, we choose the negative sample G_m randomly according to sampling distribution P_j^r [31] and replace G_j by G_m during the training process.

5 Analysis

In this section, we analyze the expressive power of PGON. We prove that our model is more powerful than other graph of graphs neural network.

5.1 Framework of PGON

The framework of PGON shown in Equation 3 can distinguish the non-isomorphic graphs. This simple structure with multiple MLP blocks can achieve great expressive power equivalents to the 3-WL test. In this subsection, we introduce how PGON implements the 2-FWL graph isomorphism test algorithm which has the equivalent expressive power as the 3-WL test algorithm.

Theorem 1 *Given two graphs $G = (V, E, x)$ and $G' = (V', E', x')$. If these two graphs can be distinguished by the 3-WL isomorphism test, there exists a graph neural network F so that $F(G) \neq F(G')$*

Proof To prove the framework of the proposed PGON has the 3-WL test expressive power, we show that the model can implement the 2-FWL algorithm which has the equivalent distinguish power as 3-WL isomorphism test algorithm.

With the given $G = (V, E, x)$, the input tensor is built as $\boldsymbol{X} \in \mathcal{R}^{n^2 \times (e+2)}$, where $\boldsymbol{X}_{:;:;e+1}$ encodes the adjacency matrix of G and $\boldsymbol{X}_{i;i;1:e}$ encodes the feature vector of each vertex $v_i \in V$, the elements outside the diagonal $\boldsymbol{X}_{j;k;1:e}, j \neq k$ are all zeros, and $\boldsymbol{X}_{:;:;e+1}$ is the identity matrix. First, we need to build the 2-FWL initialization that colors the 2-tuples by their isomorphism type. With the given adjacency matrix $\boldsymbol{A} = \boldsymbol{X}_{:;:;e+1}$, and input features $\boldsymbol{H} = \boldsymbol{X}_{:;:;1:e}$, the tensor $\boldsymbol{C} \in \mathcal{R}^{n^2 \times (4e+1)}$ is constructed using the following colors matrices. $\boldsymbol{A} \cdot \boldsymbol{H}_{:,:,j}$, $(\mathbf{1}\mathbf{1}^T - \boldsymbol{A}) \cdot \boldsymbol{H}_{:,:,j}$, $\boldsymbol{H}_{:,:,j} \cdot \boldsymbol{A}$, $\boldsymbol{H}_{:,:,j} \cdot (\mathbf{1}\mathbf{1}^T - \boldsymbol{A})$, where $j \in [e]$ and $\mathbf{1}\mathbf{1}^T - \boldsymbol{A}$ is the adjacency matrix of the complement graph. The last channel indicates whether $v_{i_1} = v_{i_2}$.

To follow the 2-FWL update steps, we perform each update step in the following way. As mentioned in Section 3.4, when the matrix $\boldsymbol{B} \in \mathcal{R}^{n \times 2a}$ is given, where $\boldsymbol{B}_{j;:} = (\boldsymbol{X}_{j;i_2;:}, \boldsymbol{X}_{i_1;j;:})$. We would like to compute the output tensor $\boldsymbol{X}^{out} \in \mathcal{R}^{n^2 \times b}$, where $\boldsymbol{X}^{out}_{i_1,i_2,:}$ has been proven in [28] that can be calculated by the network based on the MLPs.

$$\boldsymbol{X}^{out}_{i_1,i_2,l} := \sum_{j=1}^{n} m_1(\boldsymbol{X}_{j;i_2;l}) m_2(\boldsymbol{X}_{i_1;j;l}) \tag{19}$$

After this, we concatenate X and X^{out} to pair the multiset with the input color of each k-tuple.

The block introduced above could perform the 2-FWL test update. With sufficient update steps we can produce the tensor that can be used to distinguish the graphs. Our final goal is to calculate a representation tensor for each graph that indicates the histogram of its k-tuples' color. In [28], PPGN applies invariant max pooling-based operator by concatenating the max values among the diagonal and off-diagonal elements on $\boldsymbol{X}^{out}$ to produce the graph representations. The expressive power of the representations is proven both theoretically and empirically in [28].

5.2 Invariance of pooling and GIN operators

Though the operator in PPGN is invariant which can ensure expressive power, we find that simply applying the operator has a limitation in representing the importance of the nodes in the local graph. Therefore, the concatenation of max pooling on the off-diagonal elements and the self attention pooling mentioned in Equation 8 on the diagonal elements is used to produce the representations for the local graphs. Note that when the pooling ratio $\gamma = 1$, i.e., all the diagonal elements are used for the self attention pooling, our pooling operator is same as that in PPGN. Our pooling method can filter out the nodes with low importance in graph analytic tasks by adjust the pooling ratio while preserving expressive power. Then, we apply the attention-based graph isomorphic neural network to further gather information from the neighbors in the interaction graph to enhance the performance of our model. In this subsection, we prove the invariance of self-attention-based pooling and the attention-based graph isomorphic neural network which are used to calculate the graph representations. In the following, we prove the invariance of this operator $F_{GIN}(F_{pool}(\boldsymbol{X}_{out}))$.

Corollary 1 *For two mapping functions f, g, if f and g are invariant, $f \circ g$ is invariant.*

Proof Given invariant mapping f and g, by definition, we have $f(\boldsymbol{X}) = f(\boldsymbol{P}_1^T \cdot \boldsymbol{X} \cdot \boldsymbol{P}_1)$ and $g(\boldsymbol{X}) = g(\boldsymbol{P}_2^T \cdot \boldsymbol{X} \cdot \boldsymbol{P}_2)$, where $\boldsymbol{P}_i$ denotes the permutation matrix. The following result is immediate.

$$\begin{aligned} f(\boldsymbol{P}_1^T \cdot g(\boldsymbol{P}_2^T \cdot \boldsymbol{X} \cdot \boldsymbol{P}_2) \cdot \boldsymbol{P}_1) &= f(\boldsymbol{P}_1^T \cdot g(\boldsymbol{X}) \cdot \boldsymbol{P}_1) \\ &= f(g(\boldsymbol{X})) = f \circ g(\boldsymbol{X}). \end{aligned} \tag{20}$$

Therefore, we can prove the invariance of the pooling operation and GIN separately. The proof is also immediate. The pooling operation is based on the max pooling which is not sensitive to the node orders, hence, the representation for each graph is independent of the order of the nodes in the graph. The graph isomorphic neural network in Equation 9 is a message passing-based neural network which sums up the representations from the neighboring nodes with attention coefficients. It is obvious that GIN is independent of the order of nodes. Therefore, we proved the invariance of the pooling operator and attention-based GIN, which eventually infers the invariance of the calculation operator formed by these two parts.

Finally, we have proved that the PGON with the framework in Equation 3 has the expressive power equivalent to 2-FWL and 3-WL test.

6 Experiment

We evaluate our model with the graph classification and graph interaction prediction tasks on real-life datasets. In this part, we detail our experiment settings for these tasks and the results that indicate the superior performance of PGON.

6.1 Datasets

We evaluate our model on three real-life chemical interaction and protein interaction graphs. The local graphs in these three datasets are the chemicals or the proteins. The global graphs contain the interactions between the chemicals and the proteins. The datasets used are shown as follows:

- **Proteins** [1] is a benchmark graph dataset where vertices are elements within protein molecules and edges indicate that two nodes are neighbors in the amino-acid sequence or in 3D space.
- **D&D** [7] is a set of structures of enzymes and non-enzymes proteins, where nodes are amino acids, and edges represent spatial closeness between nodes.
- **MUTAG** [5] is a well-known graph classification benchmark dataset. Different from the preprocessed data in [16], we build the interaction graph with given chemicals based on the interaction score in the CCI dataset, and build more detailed molecule graphs using the description in [5].

We build the interaction graphs on the **Proteins** and **D&D** dataset using the PPI dataset. Two proteins are linked in the interaction graph if they are connected or at least have two common neighbors in the PPI network.

For the CCI and DDI prediction tasks, we use the following datasets.

- **CCI.** The CCI dataset contains the interaction probabilities between the chemical molecules. A interaction score from 0 to 999 is assigned to each interaction to indicates the probability of this interaction. Given a threshold value, we can split the datasets into CCI900 and CCI950 whose probability scores are all over 900 and 950 respectively. CCI900 has 14343 chemicals and 110078 chemical interaction edges, and CCI950 has 7606 chemicals and 34412 chemical interaction edges.
- **DDI.** DDI dataset is used for the drug adverse effect prediction experiment. The DDI dataset is originally proposed in DeepDDI [35] which contains 86 types of side effects, 1704 drugs and 191400 drug interaction edges. A vector representation $\boldsymbol{se}_r \in \mathcal{R}^{128}$ is assigned to each side effect produced by a pre-trained BERT model [6] using the name of the adverse effect.

  The molecules are transformed from the SMILE strings [41] into graphs by the open-source rdkit [19]. An initial feature vector $\boldsymbol{x}_a \in \mathcal{R}^{32}$ is assigned for every atom. The edges in the molecule graphs are weighted by the type of bonds.

6.2 Baselines and Metrics

For the graph classification task, the baseline models include graph neural network based classifiers and GNN-based approaches from the graph of graphs perspective:

- **GIN** [42] is the model that is as expressive as the Weisfeiler-Lehman graph isomorphism test.
- **PPGN** [28] is the simple and scalable GNN model that has a provable 3-WL expressive power. PPGN interleaves applications of standard Multilayer-Perceptron (MLP) applied to the feature dimension and matrix multiplication.
- **ISONN** [30] is a GNN-based model that contains two main components: graph isomorphic feature extraction component and classification component.
- **SEAL-CL** [22] is the neural network on hierarchical graphs for graph classification.
- **GoGNN** [40] is the graph neural network model built on the hierarchical graph structure, which focuses on the entity interaction prediction task.
- **DGCN** [15] is the dual graph convolutional network that learns compound representations from both the compound graphs and the inter-compound network in an end-to-end manner.

For the interaction prediction task, the following state-of-the-art baseline methods are compared:

- **DeepCCI** [18] is the CNN based model for predicting the interactions between the chemicals.
- **DeepDDI** [35] is the model designs a feature called structural similarity profile(SSP) combined with traditional MLP for DDI prediction.
- **MR-GNN** [43] is an end-to-end graph neural network with multi-resolution architecture that produces interaction between pairs of chemical graphs.
- **MLRDA** [3] is the multitask, semi-supervised model for DDI prediction.
- **SEAL-CL** [22] is the neural network on hierarchical graphs for graph classification.
- **DGCN** [15] and **GoGNN** [40].

We used the public code of the baselines and keep the settings of the models the same as discussed in the original papers. We modified the code of GoGNN and DGCN for the classification tasks. SEAL-CL is reimplemented for the interaction prediction task.

The evaluation settings in [22] are used for the classification task, and 10-fold cross validation is performed. The average accuracy is reported. For the CCI and DDI prediction task, the settings in GoGNN [40] are used to familiarize the comparison. We detail settings in the following subsections.

6.3 Classification Result

Settings. To familiarize the comparison, we divided the dataset on a 90%-10% basis for training and testing respectively. The dimensions for the graph hidden feature and

	Proteins	**D&D**	**MUTAG**
GIN	76.28%	79.91%	89.41%
PPGN	77.20%	79.97%	90.55%
ISONN	75.12%	76.28%	88.79%
SEAL-CL	76.89%	80.10%	88.17%
GoGNN	74.62%	77.69%	88.53%
DGCN	75.27%	78.94%	87.91%
PGON	78.87%	81.54%	91.15%

Table 1: Accuracy of the graph classification task.

output graph representations are set to 384 and 256 respectively. For the classification tasks, the pooling ratio is set to 1, which means all the node representations are used to produce the graph hidden feature. The average accuracy for classification is reported for evaluation.

Results. As shown in Table 1, PGON outperforms the other baseline models on all three datasets. The superior experiment results indicate that the interaction network information integrated in the PGON enhances the classification accuracy significantly compared to the models which only use the graph feature and structural information. Further, on the real-world chemical and biological dataset, the heterogeneous nature of the molecule graphs is usually ignored, and the molecule graphs are always modeled as a homogeneous graph. Our preprocess procedure assigns a simple but effective one-hot feature to each kind of atom in the molecule graphs, which also helps PGON achieve better performance. PGON also outperforms GoGNN and DGCN which shows the improvement in classification accuracy as a result of the higher expressive power of PGON.

6.4 CCI Prediction Result

Settings. As the setting in [40], CCI datasets is devided into training and testing sets with a proportion 9:1, and 10% data is randomly chosen for validation. The dimensions of local graph hidden representation, and the output local graph embedding vector are set to 384, 256, respectively. The learning rate is set to 0.01, and the pooling ratio is set to 0.5. As for the evaluation metrics, we choose *area under the ROC curve*(*AUC*) and *average precision score*(*AP*).

Results. The experiment result is shown in Table 2, PGON improves the accuracy on the CCI prediction task compared to other baseline methods. The improvement proves that PGON can capture more abundant information on multiple scales by the attribute aggregation and selection in a graph of graphs perspective compared to the baseline models which only optimize the model with molecule graph inputs pairwisely or individually. The quality of graph representations is much better with the help of the self-attention pooling layer which identifies and captures the significance of chemical subgraphs and molecular interactions. Further, the framework that follows the 2-FWL algorithm enhances the model's expressive power, and eventually improve the performance of PGON compared to GoGNN.

	CCI900		**CCI950**	
	AUC	AP	AUC	AP
DeepCCI	0.925	0.918	0.957	0.957
DeepDDI	0.891	0.886	0.916	0.915
MR-GNN	0.927	0.921	0.934	0.924
MLRDA	0.922	0.907	0.959	0.948
SEAL-CL	0.894	0.886	0.941	0.937
DGCN	0.931	0.930	0.952	0.954
GoGNN	0.937	0.932	0.963	0.962
PGON	0.939	0.936	0.968	0.965

Table 2: Result of chemical-chemical interaction prediction task.

	DDI	
	AUC	AP
DeepCCI	0.862	0.856
DeepDDI	0.915	0.912
MR-GNN	0.932	0.922
MLRDA	0.931	0.926
SEAL-CL	0.925	0.921
DGCN	0.924	0.919
GoGNN	0.943	0.933
PGON	0.946	0.937

Table 3: Result of drug-drug interaction prediction task.

6.5 DDI Prediction Result

Settings. Following the previous study, the DDI dataset is divided for training, testing, validation with ratio 6:2:2. The dimensions of local graph hidden representation, and the output local graph embedding vector are set to 384, 256, respectively. The learning rate is set to 0.001, and the pooling ratio is set to 0.5. We choose *AUC* and *average precision*(*AP*) as the evaluation metrics.

Results. Table 3 shows the experimental results for the adverse side effect prediction task. It is illustrated by the results that PGON outperforms the baseline methods with regard to the performance of drug-drug interaction prediction task. The improvement is attributed to the abundant information brought by the graph of graphs architecture and edge-filtered aggregation.

6.6 Ablation Experiment

In order to determine the effectiveness of graph of graphs framework and the expression power of the model. We conduct the ablation experiments which replace the component of our proposed model with other conventional components. The following variants of the PGON are tested for the ablation experiment.

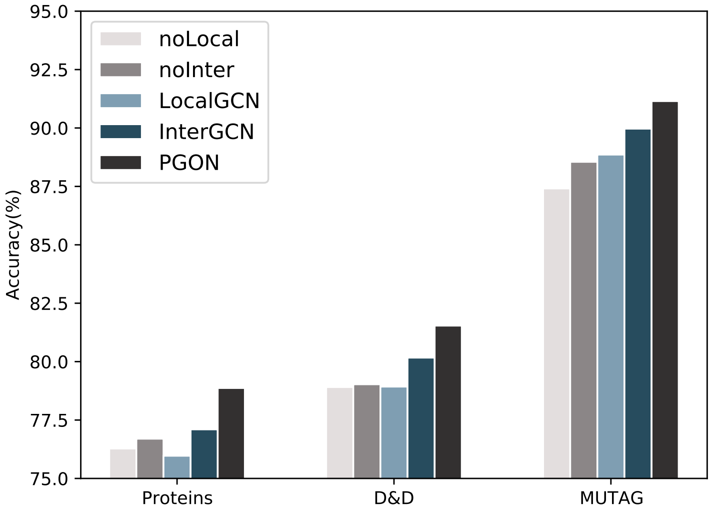


Fig. 3: Ablation experiment result for graph classification.

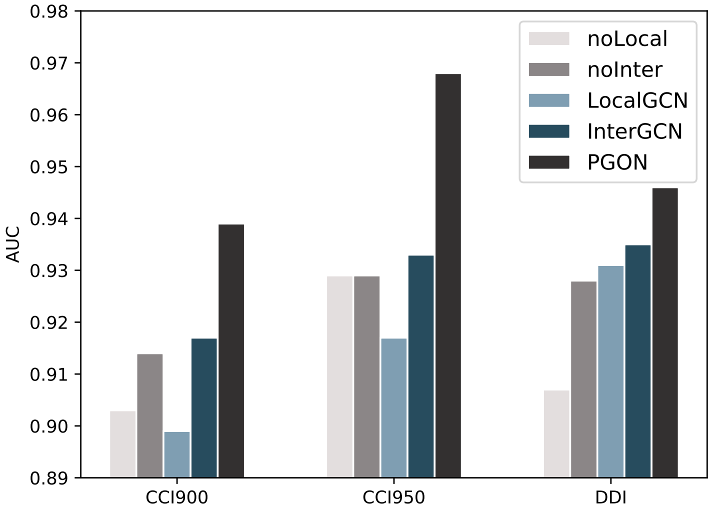
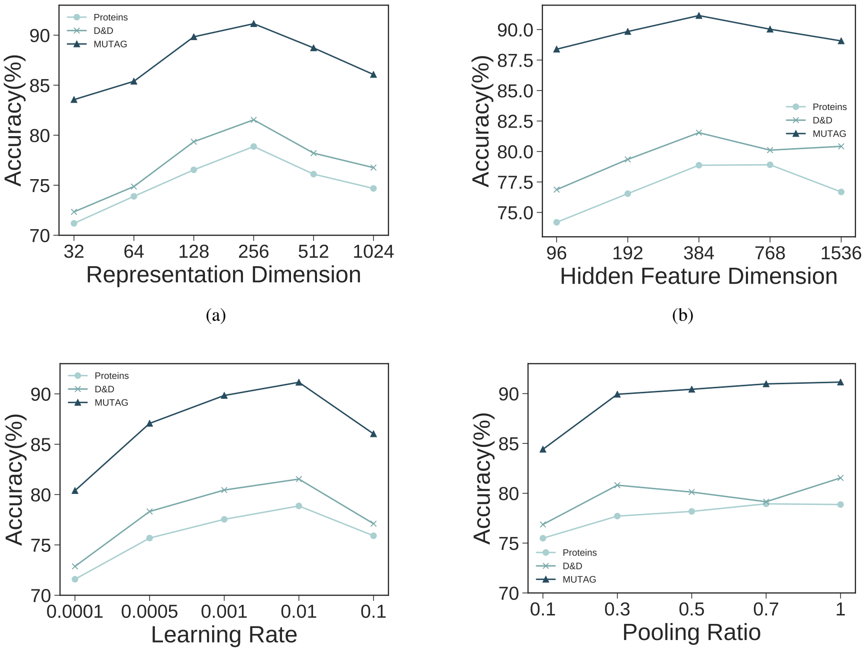


Fig. 4: Ablation experiment result for graph interaction prediction.

noLocal is the variant that only applies the attention-based graph isomorphism network on the graph interaction network while skipping the representation learning for the local graphs.

noInter is the variant that only learns the representations for local graphs while ignoring the interaction graph between the local graphs.



(a)

(b)

(c)

(d)

Fig. 5: Parameter sensitivity experiment results for graph classification

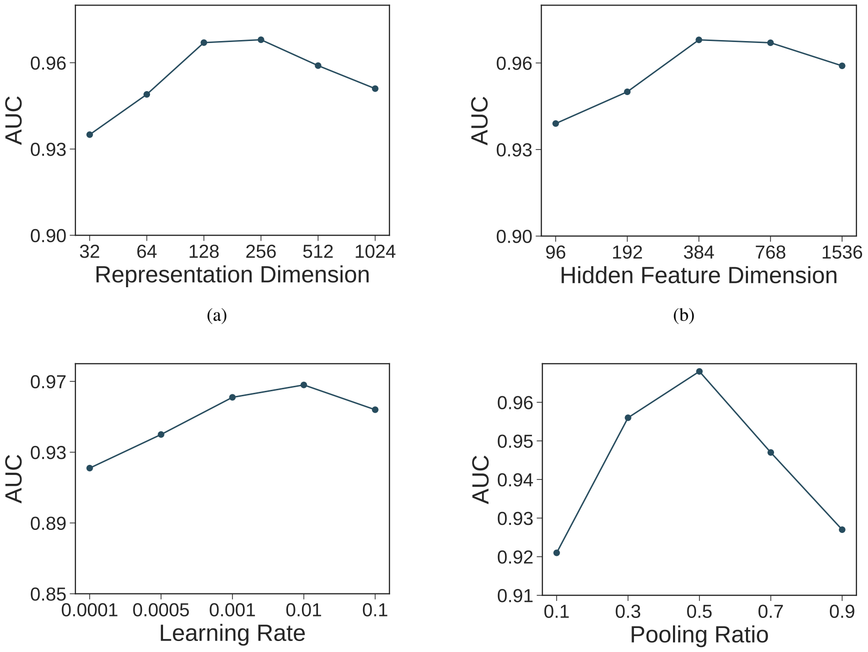
LocalGCN denotes the variant that replace the local graph neural network with conventional graph convolutional neural network [17].
InterGCN denotes the variant that uses GCN for the message passing between the graphs on the interaction network.

The results are shown in Fig. 3 and 4, which prove the effectiveness of the graph of graphs architecture and the expressive power of the entire model. Among all the variants, LocalGCN and noLocal have the most significant performance gaps between PGON, which indicates that the expressive power of PGON is the key factor in improving the performance for both classification and prediction tasks.

6.7 Parameter Sensitivity Analysis

Parameter sensitivity is analyzed in the experiment. Hidden feature dimensions, final representation dimensions, pooling ratio and learning rate are tested for both graph classification and link prediction tasks to demonstrate how the parameters influence the performance of the model. For the graph classification task, we test different parameter settings of parameters on all three datasets, while for graph interaction prediction task, the parameters are tested on the CCI950 dataset.

The results are summarized in Fig. 5 and 6. Overall, the impact of the hyperparameter variations is insignificant. The results indicate the salient point for each hyper-parameter. Therefore, the best settings for the hyper-parameters are selected as mentioned in Section 6.3, 6.4 and 6.5.



(a)

(b)

(c)

(d)

Fig. 6: Parameter sensitivity experiment results for graph interaction prediction

7 Conclusion

In this paper, we propose the graph neural network model PGON that can be applied to both graph classification and graph interaction prediction tasks. PGON is able to preserve the structure and feature information of the graph of graphs, which empirically enhances the performance on link prediction and graph classification. PGON also has the provably great expressive power which helps the model to distinguish the structure of different graphs. The extensive experiments show the effectiveness of PGON on multiple graph analysis tasks.

References

1. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H.: Protein function prediction via graph kernels. In: Proceedings Thirteenth International Conference on Intelligent Systems for Molecular Biology 2005, Detroit, MI, USA, 25-29 June 2005, pp. 47–56 (2005)
2. Cai, J.Y., Fürer, M., Immerman, N.: An optimal lower bound on the number of variables for graph identification. Combinatorica **12**(4), 389–410 (1992)
3. Chu, X., Lin, Y., Wang, Y., Wang, L., Wang, J., Gao, J.: Mlrda: a multi-task semi-supervised learning framework for drug-drug interaction prediction. In: IJCAI (2019)
4. D'Agostino, G., Scala, A.: Networks of networks: the last frontier of complexity, vol. 340. Springer (2014)
5. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry **34**(2), 786–797 (1991)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Dobson, P.D., Doig, A.J.: Distinguishing enzyme structures from non-enzymes without alignments. Journal of molecular biology **330**(4), 771–783 (2003)
8. Dong, G., Gao, J., Du, R., Tian, L., Stanley, H.E., Havlin, S.: Robustness of network of networks under targeted attack. Physical Review E **87**(5), 052804 (2013)
9. Errica, F., Podda, M., Bacciu, D., Micheli, A.: A fair comparison of graph neural networks for graph classification. arXiv preprint arXiv:1912.09893 (2019)
10. Fey, M., Lenssen, J.E., Morris, C., Masci, J., Kriege, N.M.: Deep graph matching consensus. In: International Conference on Learning Representations (2019)
11. Grohe, M.: Descriptive complexity, canonisation, and definable graph structure theory, vol. 47. Cambridge University Press (2017)
12. Grohe, M., Otto, M.: Pebble games and linear equations. The Journal of Symbolic Logic pp. 797–844 (2015)
13. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS, pp. 1024–1034 (2017)
14. Hao, Y., Cao, X., Fang, Y., Xie, X., Wang, S.: Inductive link prediction for nodes having only attribute information. arXiv preprint arXiv:2007.08053 (2020)
15. Harada, S., Akita, H., Tsubaki, M., Baba, Y., Takigawa, I., Yamanishi, Y., Kashima, H.: Dual graph convolutional neural network for predicting chemical networks. BMC bioinformatics **21**, 1–13 (2020)
16. Kersting, K., Kriege, N.M., Morris, C., Mutzel, P., Neumann, M.: Benchmark data sets for graph kernels (2016). URL `http://graphkernels.cs.tu-dortmund.de`
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR 2017 (2017)
18. Kwon, S., Yoon, S.: Deepcci: End-to-end deep learning for chemical-chemical interaction prediction. In: Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics, pp. 203–212 (2017)
19. Landrum, G.: Rdkit documentation. Release **1**, 1–79 (2013)
20. Lee, J., Lee, I., Kang, J.: Self-attention graph pooling. arXiv preprint arXiv:1904.08082 (2019)
21. Lee, J.B., Rossi, R., Kong, X.: Graph classification using structural attention. In: SIGKDD, pp. 1666–1674. ACM (2018)
22. Li, J., Rong, Y., Cheng, H., Meng, H., Huang, W., Huang, J.: Semi-supervised graph classification: A hierarchical graph perspective. In: L. Liu, R.W. White, A. Mantrach, F. Silvestri, J.J. McAuley, R. Baeza-Yates, L. Zia (eds.) The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pp. 972–982. ACM (2019)
23. Li, Y., Gu, C., Dullien, T., Vinyals, O., Kohli, P.: Graph matching networks for learning the similarity of graph structured objects. arXiv preprint arXiv:1904.12787 (2019)
24. Lian, D., Liu, Q., Chen, E.: Personalized ranking with importance sampling. In: WWW 2020, p. 1093–1103 (2020)
25. Lian, D., Wang, H., Liu, Z., Lian, J., Chen, E., Xie, X.: Lightrec: A memory and search-efficient recommender system. In: WWW 2020, p. 695–705 (2020). DOI 10.1145/3366423.3380151. URL `https://doi.org/10.1145/3366423.3380151`
26. Liao, R., Li, Y., Song, Y., Wang, S., Hamilton, W., Duvenaud, D.K., Urtasun, R., Zemel, R.: Efficient graph generation with graph recurrent attention networks. In: Advances in Neural Information Processing Systems, pp. 4255–4265 (2019)

27. Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L., Qi, Y.: Geniepath: Graph neural networks with adaptive receptive paths. In: AAAI, vol. 33, pp. 4424–4431 (2019)
28. Maron, H., Ben-Hamu, H., Serviansky, H., Lipman, Y.: Provably powerful graph networks. In: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.B. Fox, R. Garnett (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 2153–2164 (2019)
29. Menche, J., Sharma, A., Kitsak, M., Ghiassian, S.D., Vidal, M., Loscalzo, J., Barabási, A.L.: Uncovering disease-disease relationships through the incomplete interactome. Science **347**(6224) (2015)
30. Meng, L., Zhang, J.: Isonn: Isomorphic neural network for graph representation learning and classification. CoRR **abs/1907.09495** (2019)
31. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
32. Nguyen, T.D., Phung, D., et al.: Unsupervised universal self-attention network for graph classification (2019)
33. Ni, J., Tong, H., Fan, W., Zhang, X.: Inside the atoms: ranking on a network of networks. In: SIGKDD, pp. 1356–1365 (2014)
34. Rome, E., Langeslag, P., Usov, A.: Federated modelling and simulation for critical infrastructure protection. In: Networks of networks: the last frontier of complexity, pp. 225–253. Springer (2014)
35. Ryu, J.Y., Kim, H.U., Lee, S.Y.: Deep learning improves prediction of drug–drug and drug–food interactions. Proceedings of the National Academy of Sciences **115**(18), E4304–E4311 (2018)
36. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference, pp. 593–607. Springer (2018)
37. Shi, H., Fan, H., Kwok, J.T.: Effective decoding in graph auto-encoder using triadic closure. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 906–913 (2020)
38. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
39. Wang, H., Lian, D., Zhang, Y., Qin, L., He, X., Lin, Y., Lin, X.: Binarized graph neural network. World Wide Web pp. 1–24 (2021)
40. Wang, H., Lian, D., Zhang, Y., Qin, L., Lin, X.: Gognn: Graph of graphs neural network for predicting structured entity interactions. arXiv preprint arXiv:2005.05537 (2020)
41. Weininger, D., Weininger, A., Weininger, J.L.: Smiles. 2. algorithm for generation of unique smiles notation. Journal of chemical information and computer sciences **29**(2), 97–101 (1989)
42. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019)
43. Xu, N., Wang, P., Chen, L., Tao, J., Zhao, J.: Mr-gnn: Multi-resolution and dual graph neural network for predicting structured entity interactions. arXiv preprint arXiv:1905.09558 (2019)
44. Yang, C., Zhuang, P., Shi, W., Luu, A., Li, P.: Conditional structure generation through graph variational generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 1340–1351 (2019)
45. You, J., Liu, B., Ying, Z., Pande, V., Leskovec, J.: Graph convolutional policy network for goal-directed molecular graph generation. In: Advances in neural information processing systems, pp. 6410–6421 (2018)
46. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: NIPS, pp. 5165–5175 (2018)
47. Zhang, M., Cui, Z., Neumann, M., Chen, Y.: An end-to-end deep learning architecture for graph classification. In: AAAI (2018)
48. Zhang, Z., Bu, J., Ester, M., Zhang, J., Yao, C., Yu, Z., Wang, C.: Hierarchical graph pooling with structure learning. arXiv preprint arXiv:1911.05954 (2019)
49. Zitnik, M., Agrawal, M., Leskovec, J.: Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics **34**(13), i457–i466 (2018)