KG-MTL: Knowledge Graph Enhanced Multi-Task Learning for Molecular Interaction

Tengfei Ma, Xuan Lin*, Bosheng Song, Philip S. Yu, *Fellow, IEEE*, Xiangxiang Zeng*, *Senior Member, IEEE*

Abstract—Molecular interaction prediction is essential in various applications including drug discovery and material science. The problem becomes quite challenging when the interaction is represented by unmapped relationships in molecular networks, namely molecular interaction, because it easily suffers from (i) insufficient labeled data with many false-positive samples, and (ii) ignoring a large number of biological entities with rich information in the knowledge graph. Most of the existing methods cannot properly exploit the information of knowledge graph and molecule graph simultaneously. In this paper, we propose a large-scale **K**nowledge **G**raph enhanced **M**ulti-**T**ask **L**earning model, namely KG-MTL, which extracts the features from both knowledge graph and molecular graph in a synergistic way. Moreover, we design an effective *Shared Unit* that helps the model to jointly preserve the semantic relations of drug entity and the neighbor structures of the compound in both knowledge graph and molecular graph. Extensive experiments on four real-world datasets demonstrate that our proposed KG-MTL outperforms the state-of-the-art methods on two representative molecular interaction prediction tasks: drug-target interaction prediction and compound-protein interaction prediction. The source code of KG-MTL is available at https://github.com/xzenglab/KG-MTL.

Index Terms—Machine Learning, Knowledge Graph, Multi-Task Learning, Drug Discovery

1 Introduction

Molecular interaction prediction between targets plays a critical role in many applications, including pharmacology and clinical application [1]. Such a process is to predict the unmapped relationships between unknown targets, namely molecular interaction prediction (MIP), and it is one of the fundamental steps to explore the candidate drugs for targets in drug discovery, which further speeds up the costly and time-consuming process of experiment [2], [3]. A typical MIP pipeline takes the features of drug and target (e.g., protein or gene) as the input and outputs the interaction probability of given drug-target pair. The predicted interactions are beneficial to various subsequent tasks, including molecular property prediction [4], [5], [6], [7], drug reactions [1], [8], drug effectiveness [9] and drug side effects prediction [10], [11]. However, accurately recognizing the molecular interaction with computational methods remains challenging.

Previous approaches on molecular interaction have exploited various types of molecular features, such as chemical structures [12] and the similarities between drug-target pairs [13]. However, these methods heavily depend on the design of hand-crafted features and domain knowledge from labeled data. Recently, various deep neural networks and graph neural networks have been developed and achieved excellent performance for MIP. Most existing methods focus on modeling each chemical molecule as the molecular graph to capture the neighbor structure information [14], [15], [16], or integrating various networks as side information to boost the prediction performance, including protein interactions [17], drug-drug interactions [18], and drug-target interactions [19], [20]. However, these works are using either local features or a relatively small network that can not comprehensively consider most biological entities with comparison to large-scale knowledge graphs (e.g., DRKG includes 97,238 entities and 5,874,261 triples). Furthermore, there are many false positives (i.e., samples originally regarded as positive are actually potential negative ones) and limited labeled samples in the constructed networks, which will result in negative influence on model performance [21], [22].

Recent studies adopted knowledge graph (KG) to enhance the biological data reliability in downstream tasks, such as drug-drug interaction (DDI) prediction [23], adverse DDI [24], and unknown drug-target interaction (DTI) or compound-protein interaction (CPI) prediction [25]. They apply knowledge graph representation learning to integrate multiple data sources. However, these works directly learn latent entity embedding without considering multiple relationships, which are limited in mining semantic relations and topological structures of each entity in KG. For example, KGNN [23] merely focused on the DDI information while it ignored other types of entities and relations in KG.

We observe that existing methods on molecular interaction prediction do not make full use of knowledge graph as well as molecular graph and only consider partial information. These limitations and the success of multi-task learn-

- *T. Ma, B. Song and X.zeng were with the School of Computer Science and Engineering, Hunan University, Changsha 410012, China. E-mail: see {tfma, boshengsong, xzeng}@hnu.edu.cn*
- *X. Lin was with the School of Computer Science, Xiangtan University, Xiangtan 411105, China, and also with the Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education (Xiangtan University), Xiangtan 411105, China. E-mail: see jack_lin@xtu.edu.cn*
- *Philip S. Yu was with the Department of Computer Science, University of Illinois at Chicago. E-mail: psyu@cs.uic.edu*
- ** Corresponding authors*

Manuscript received XXX; revised XXX

ing [26], [27], [28], [29] motivate us to develop a new method to fully exploit the information from both knowledge graph and molecular graph to predict the molecular interaction. In particular, we propose a novel large-scale knowledge graph enhanced multi-task learning model, named KG-MTL. The idea of KG-MTL is natural and intuitive, which combines the topological structure of the molecular graph and the corresponding biological entities of KG, by using multi-task learning strategies. In addition, we adopted a comprehensive biological KG including drugs, diseases, proteins, genes, pathways, and expression. Therefore, we can mine a large number of potential drug-target interactions from the KG that can improve the performance of other tasks by some query patterns (see details in **Section 3.7**). In a nutshell, our framework consists of three major modules. Specifically, (i) *DTI module* is used to extract the features of drugs and related entities from large-scale KG. (ii) *CPI module* is adopted to learn two representations of the molecular graph and protein sequences. (iii) *Shared Unit* is designed to share task-independent drug features between the previous two modules, by combining the molecular representation of compound and corresponding drug entity embedding from KG. In summary, the contributions of this work are as follows:

1) To the best of our knowledge, this is the first work to apply a large-scale knowledge graph on a multi-task learning model, namely KG-MTL, to the problem of molecular interaction prediction.
2) The proposed KG-MTL has two distinct technical highlights. (i) KG-MTL jointly extracts the features from both knowledge graph and molecule graph synergistically; and (ii) the novel shared unit is designed to capture the semantic relations of drug entity in the knowledge graph while preserving the topological structures of the compound within the molecular graph.
3) Extensive experiments on four real-world datasets illustrate that KG-MTL outperforms the state-of-the-art molecular interaction prediction baselines in two representative applications: drug-target interaction prediction and compound-protein interaction prediction.

The rest of this paper is organized as follows. In Section 2, graph-based and KG-based methods on the MIP prediction tasks are introduced. The formulation and the details of KG-MTL are presented in Section 3. Section 4 illustrates various experiments (e.g., Ablation Experiments) to validate the effectiveness of KG-MTL. And in Section 5, we discuss the future work and existing issues to improve the molecular interaction prediction.

2 Related Works

Over the years, molecular interaction prediction (MIP) has received great attention in drug discovery. Previous works mainly focused on investigating various types of molecular features to predict the molecular interaction. For example, a bipartite local model was proposed to predict unknown targets by using chemical structures information [12]. And Gaussian interaction profile kernels were designed to describe the similarities among drug-target interaction profiles [13]. However, these methods heavily depend on feature engineering and domain knowledge.

2.1 Graph-based Methods.

More recently, various deep neural networks and graph neural networks (GNNs) have achieved excellent performance for molecular interaction prediction. In particular, an end-to-end deep learning framework named GNN-CPI [15] that applied GNN layer to extract the fingerprint features of the compound represented by molecular graph. In the same line of work, a novel heterogeneous network named NeoDTI [19] learned low dimensional vector representation of drug by integrating multiple drug-related networks to predict the unknown target. Moreover, MONN was proposed to jointly predict both non-covalent interactions and binding affinities between compounds and proteins [30]. However, these methods are either local features of the molecule or relatively small to consider most biological entities. With comparison to the graph-based (a.k.a network-based) methods, our proposed KG-MTL can automatically extract the features of drug from molecular graph, and also obtain the semantic relations information between drug and other entities from the large-scale knowledge graph.

2.2 KG-based Methods.

Recent studies on molecular interaction prediction also apply large-scale knowledge graph (KG) to extract various biological entities. For example, a novel method named GAMENet was constructed to integrate multiple datasets with DDI information in KG to predict unknown adverse DDI [24]. And TriModel adopted KG embedding to learn the representations of drug and target for DTI prediction [25]. These models usually extract drug features using various embedding methods, and directly learn entity embedding from KG, while they easily ignore the semantic relations and topological features between drug and other entities. Compared with this line of methods, our KG-MTL differs from them in the following aspects: (i) our proposed framework jointly considers multiple types of drug entity and relations from knowledge graph and the neighbor structures information from the molecular graph, to further improve the performance between two tasks. and (ii) we develop an effective shared unit module to train the two tasks that works well under our framework by synergistically using multi-task learning strategies.

3 Method

In this section, we firstly formulate molecular interaction prediction problem. Then we introduce the framework of the proposed KG-MTL. Finally, we discuss the model training and learning strategy in detail.

3.1 Preliminaries

Problem Definition. For ease of understanding of our proposed method, in this paper, we focus on two representative applications of molecular interaction prediction: drug-target interaction (DTI) prediction and compound-protein interaction (CPI) prediction. In DTI task, we aim to estimate the interaction probability p_{ij}^{dti} of a drug-target pair (d_i, t_j) in *knowledge graph* G. As to CPI task, our goal is to evaluate the occurrence probability score p_{ij}^{cpi} with a compound-protein pair (g_i, s_j) in *molecule graph*. Therefore, we aim to learn a

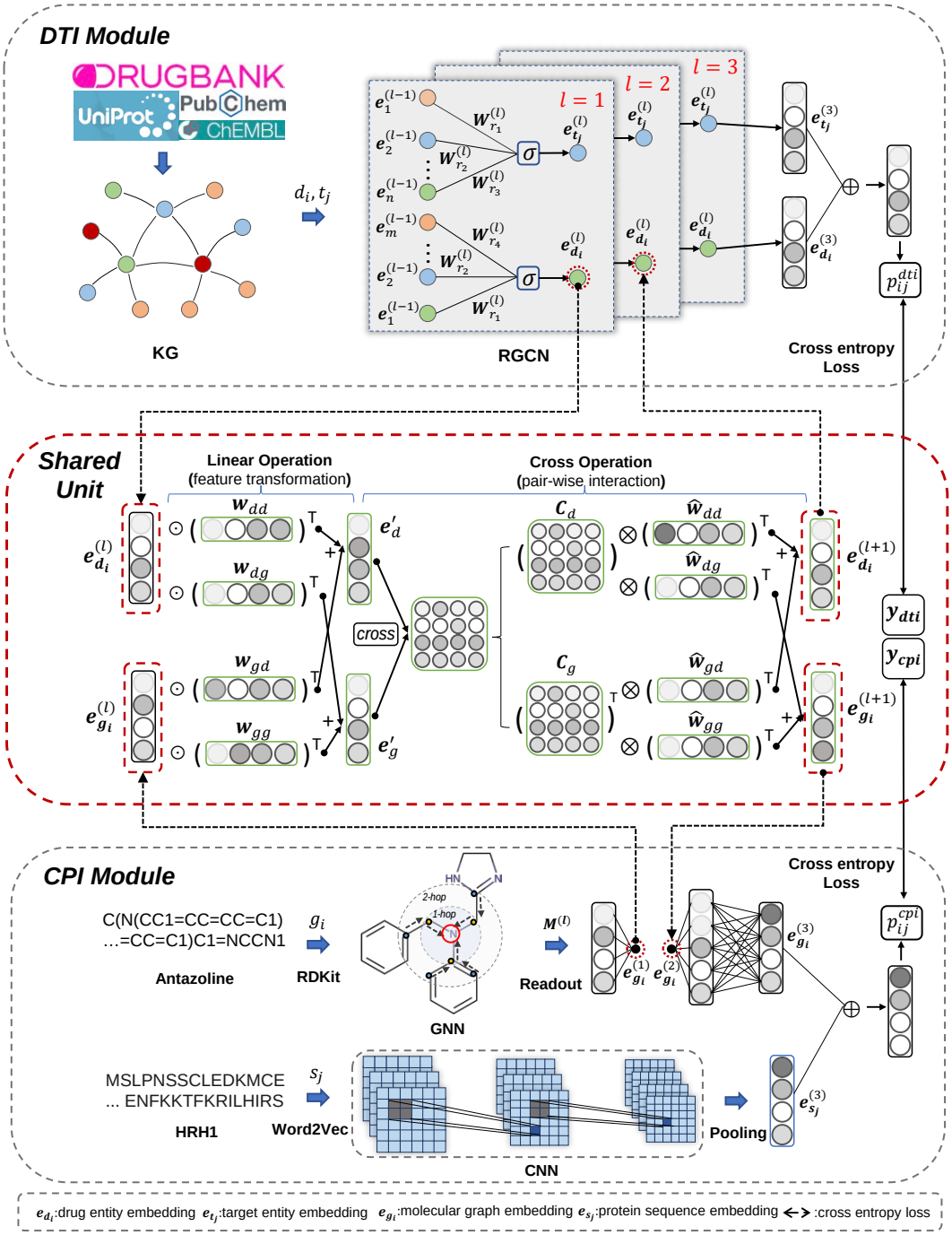


Fig. 1. The framework of our proposed KG-MTL.

prediction function $(p_{ij}^{dti}, p_{ij}^{cpi}) = \mathcal{F}((d_i, t_j), (g_i, s_j)|\Theta, G)$, where Θ denotes the model parameters.

Knowledge Graph. We consider a KG as G that provides base information for a drug-target pair $(d_i, t_j) \in P_{dti}$ in DTI task, where P_{dti} is the set of DTI pairs. And we define $\mathbf{e}_{d_i}$ and $\mathbf{e}_{t_j}$ as the learned embeddings of corresponding drug entity d_i and target entity t_j from G, respectively.

Molecule Graph. Given compound-protein pair $(g_i, s_j) \in P_{cpi}$, where P_{cpi} denotes the set of CPI pairs, the compound g_i is defined by a molecule graph transformed from SMILES using RDKit [31]. And $g_i = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ denotes the set of atoms and $\mathcal{E}$ is the set of edges between atoms. Then we denote a global embedding of molecule graph g_i as $\mathbf{e}_{g_i}$. Meanwhile, we define a protein s_j in the format of amino acid sequences. And we represent the protein sequence embedding as $\mathbf{e}_{s_j}$ by using word embedding.

3.2 Framework of KG-MTL

The framework of KG-MTL is illustrated in Fig. 1, and it consists of three modules. In *DTI module*, the relational graph convolutional network (RGCN) is applied to learn the semantic relations and topological structure information of drug and target entities from the knowledge graph,

which helps to predict unknown drug-target interaction. In *CPI module*, we adopt convolutional neural network (CNN) and graph convolutional network (GCN) to extract more chemical contexts and the molecular structures from protein sequence and compound molecular graph respectively. And more importantly, we design an effective *Shared Unit* to fuse the molecular structure of compound with the semantic relations of the corresponding drug entity from the previous two modules, to further improve the model performance.

3.2.1 DTI Module.

In the DTI task, we learn latent representations of drug and target entities from large-scale DRKG [32] which provides much information shared for two tasks as shown in Fig. 1 (i.e., DTI Module). Specifically, we first generate a subgraph from DRKG using *neighbor sampling* [33] to make the model converge faster. Then we employ a 3-layer RGCN model [34] to extract the semantic relations and topological structure of entities from the previously generated subgraph. And multiple aggregation techniques are adopted in RGCN to consider different types of relationships between entities, and the specific operations are as follows:

$$\mathbf{e}_i^{(l+1)} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}\mathbf{W}_r^{(l)}\mathbf{e}_j^{(l)} + \mathbf{W}_0^{(l)}\mathbf{e}_i^{(l)}\Big), \tag{1}$$

where $\mathbf{e}_i^{(l)}$ and $\mathbf{e}_j^{(l)}$ denote the embedding of i^{th} and j^{th} entity in l^{th} RGCN layer (e.g., $\mathbf{e}_{d_i}^{(l)}$ and $\mathbf{e}_{t_j}^{(l)} \in \mathbb{R}^{dim}$), and $\mathbf{W}_r$ represents the weights of relation r, and $\mathcal{N}_i^r$ denotes the set of neighbors of node entity i under relation $r \in \mathcal{R}$, and $c_{i,r}$ is a normalization constant, where we set $c_{i,r} = |\mathcal{N}_i^r|$, σ denotes the activation function (i.e., ReLU). Subsequently, we combine the embeddings of drug and target entities that learned from the last RGCN layer and termed it as a concatenated vector $[\mathbf{e}_{d_i}^{(3)}; \mathbf{e}_{t_j}^{(3)}]$. Finally, we input the concatenated vector into a classifier, which consists of a multi-layer perception (MLP) and a *sigmoid* layer to output the interaction probability p_{ij}^{dti} of given DTI pair.

3.2.2 CPI Module.

As shown in Fig. 1 (i.e., CPI Module), given a molecule graph g_i and a protein sequence s_j in CPI pair (g_i, s_j), we first use a GCN layer to continuously update the node embedding $\boldsymbol{v_i}$ in a molecular graph through message passing [35], where each atom node $v_i \in \mathcal{V}$ is the i-th atom initialized by a 78-dimensional feature vector $\boldsymbol{v_i}$ [36]. Next we input the hidden feature matrix $\boldsymbol{M}^{(l)}$ of the last GCN layer into a MLP readout layer, to obtain the global representation $\mathbf{e}_{g_i}^{(1)}$ and we have:

$$\mathbf{e}_{g_i}^{(1)} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\sigma(f(\boldsymbol{v}_i)). \tag{2}$$

3.2.3 Shared Unit

To effectively associate DTI and CPI modules and to address the limitations in previous works, we design a novel *Shared Unit* to mix the molecular structures of compound with the semantic relations of the corresponding drug entity in the knowledge graph. As shown in Fig. 1 (i.e., Shared Unit), given drug entity d_i and the corresponding compound g_i, we firstly take the drug entity embedding $\mathbf{e}_{d_i}^{(l)}$ of the l^{th} RGCN layer from DTI module and molecular graph embedding $\mathbf{e}_{g_i}^{(l)}$ of the l^{th} linear layer from CPI module as the input of *Shared Unit* ($\mathbf{e}_{d_i}^{(l)}$, $\mathbf{e}_{g_i}^{(l)} \in \mathbb{R}^{dim}$). Secondly, we utilize four trainable weights ($\boldsymbol{w}_{dd}, \boldsymbol{w}_{dg}, \boldsymbol{w}_{gg}, \boldsymbol{w}_{gd} \in \mathbb{R}^{dim}$) to automatically learn the weight of each input feature as follows:

$$\mathbf{e}_d^{'} = \boldsymbol{w}_{dd}^{\mathrm{T}} \odot \mathbf{e}_{d_i}^{(l)} + \boldsymbol{w}_{gd}^{\mathrm{T}} \odot \mathbf{e}_{g_i}^{(l)}, \tag{3}$$

$$\mathbf{e}_g^{'} = \boldsymbol{w}_{gg}^{\mathrm{T}} \odot \mathbf{e}_{g_i}^{(l)} + \boldsymbol{w}_{dg}^{\mathrm{T}} \odot \mathbf{e}_{d_i}^{(l)}, \tag{4}$$

where $\mathbf{e}_d^{'}$, $\mathbf{e}_g^{'} \in \mathbb{R}^{dim}$ are the features obtained from linear transformation of $\mathbf{e}_{d_i}^{(l)}$ and $\mathbf{e}_{g_i}^{(l)}$ respectively, and $\odot$ denotes the element-wise multiplication (i.e., linear operation). Thirdly, to further combine with the feature vectors of drug and compound, we construct a cross matrix $\boldsymbol{C} \in \mathbb{R}^{dim\times dim}$ by pairwise interactions of their latent feature $\mathbf{e}_d^{'}$ and $\mathbf{e}_g^{'}$ (i.e., cross operation) as shown in Eq. (5).

$$\boldsymbol{C} = \mathbf{e}_d^{'}(\mathbf{e}_g^{'})^{\mathrm{T}} = \begin{bmatrix} \mathbf{e}_d^{'(1)}\mathbf{e}_g^{'(1)} & \cdots & \mathbf{e}_d^{'(1)}\mathbf{e}_g^{'(dim)} \\ \vdots & \vdots & \vdots \\ \mathbf{e}_d^{'(dim)}\mathbf{e}_g^{'(1)} & \cdots & \mathbf{e}_d^{'(dim)}\mathbf{e}_g^{'(dim)} \end{bmatrix}. \tag{5}$$

To maintain the symmetry of learned embeddings, we mix the features along both horizontal and vertical directions by designing two intermediate variables $\boldsymbol{C}_d = \boldsymbol{C}$ and $\boldsymbol{C}_g = \boldsymbol{C}^{\mathrm{T}}$, where $\boldsymbol{C}_d$, $\boldsymbol{C}_g \in \mathbb{R}^{dim\times dim}$, so we can capture the high-dimensional features of drug entity and compound molecule. Finally, we input $\boldsymbol{C}_d$ and $\boldsymbol{C}_g$ into a non-linear operator to project them back to the original feature space of the input of two modules, and they are calculated as follows:

$$\mathbf{e}_{d_i}^{(l+1)} = \boldsymbol{C}_d \otimes \hat{\boldsymbol{w}}_{dd} + \boldsymbol{C}_g \otimes \hat{\boldsymbol{w}}_{gd} + \boldsymbol{b}_d, \tag{6}$$

$$\mathbf{e}_{g_i}^{(l+1)} = \boldsymbol{C}_d \otimes \hat{\boldsymbol{w}}_{gg} + \boldsymbol{C}_g \otimes \hat{\boldsymbol{w}}_{dg} + \boldsymbol{b}_g, \tag{7}$$

where $\otimes$ denotes matrix multiplication, $\hat{\boldsymbol{w}}_{dd}$, $\hat{\boldsymbol{w}}_{gd}$, $\hat{\boldsymbol{w}}_{dg}$, $\hat{\boldsymbol{w}}_{gg}$ are trainable weights and $\boldsymbol{b}_d$, $\boldsymbol{b}_g$ represent bias vectors. $\mathbf{e}_{d_i}^{(l+1)}$ and $\mathbf{e}_{g_i}^{(l+1)}$ are used as the inputs of next layer in DTI and CPI modules. **Note that** once the *Shared Unit* is used, it will merge the learned embeddings of the drug entity and the corresponding compound into a new representation, which will update the drug or compound representation of the candidate layer in each module for iterative training respectively. Otherwise, it goes directly to the next layer for model training. Actually, the *Shared Unit* can be regarded as a part of the component that added between the linear and RGCN layer. Here, we only added a *Shared Unit* in the first component (i.e., the first linear and RGCN layer) and it can be added in the second and subsequent component as the number of layers increases. Different from the traditional operation method in multi-task learning (e.g., Cross-Stitch [26]), our proposed method can obtain more high-order features from knowledge graph, we suggest that *Shared Unit* should be modeled at the lower layer to capture more general features. We will evaluate the impact of number and settings of *Shared Unit* in parameter sensitivity analysis (Section 4.8).

3.3 Design Decisions for Shared Unit

We design the novel *Shared Unit* for multi-task learning between related tasks. The primary idea is to apply the

explicit features efficiently crossing different tasks. For the sake of simplicity, we use multi-task learning with two tasks in this work. Intuitively, the designed *Shared Unit* can regularize both tasks by learning and enforcing the shared representations (i.e., crossing features). And the proposed *Shared Unit* is composed of two parts, including linear and cross operation.

Linear Operation. The linear operation can model the linear combinations of features from both tasks using dot product with learnable parameters. Intuitively, we regard it as an *attention mechanism* [37], where the importance (i.e., weight) of different feature dimensions can be learned to improve the representation ability of drug and compound features.

Cross Operation. Considering the Weierstrass approximation theorem [38], any functions under certain smoothness assumption can be approximated by polynomial to arbitrary accuracy. So we examine the ability of high-order interaction approximation of the *Shared Unit*.

Theorem 1. *Denote the representations of given drug and compound entities as* $\mathbf{e}_d^{'} = [e_d^{'(1)} \cdots e_d^{'(dim)}]^T$ *and* $\mathbf{e}_g^{'} = [e_g^{'(1)} \cdots e_g^{'(dim)}]^T$*, respectively. Then, the cross terms* $\mathbf{e}_d^{'}$ *and* $\mathbf{e}_g^{'}$ *in* $\|\mathbf{e}_{d(L)}^{'}\|_1$ *and* $\|\mathbf{e}_{g(L)}^{'}\|_1$ *(i.e., the L1-norm of* $\mathbf{e}_d^{'}$ *and* $\mathbf{e}_g^{'}$*) with maximal degree are represented by* $k_{\alpha,\beta} e_d^{'(1)\alpha_1} \cdots e_d^{'(dim)\alpha_{dim}} e_g^{'(1)\beta_1} \cdots e_g^{'(dim)\beta_{dim}}$*, where* $k_{\alpha,\beta} \in \mathbb{R}, \alpha_i, \beta_i \in \mathbb{N}$ *for* $i \in \{1, \cdots, dim\}, \alpha_1 + \cdots + \alpha_{dim} = 2^{L-1}$*, and* $\beta_1 + \cdots + \beta_{dim} = 2^{L-1}$ *(*$L \geq 1$*).*

And the $\prod_{i=1}^{dim} e_d^{'(i)\alpha_i} e_g^{'(i)\beta_i}$ is also called combinatorial feature, which can be obtained by measuring the interactions of various original features. The *Shared Unit* can automatically model the high-level fused representation of drugs and compounds according to Theorem 1, which proves the superior approximation ability of the cross operation. Moreover, we empirically evaluate each operation in the **Ablation Experiments** (Section 4.7).

3.4 Model Training

Given the DTI pairs, CPI pairs and the corresponding labels in the training set for both two tasks, our optimization goal is to minimize the following cross-entropy loss as follows:

$$\mathcal{L}_{dti} = -\sum_{(d_i,t_j)\in\mathcal{P}_{dti}} y_{ij}^{dti} \log p_{ij}^{dti} + (1 - y_{ij}^{dti}) \log(1 - p_{ij}^{dti}), \tag{8}$$

$$\mathcal{L}_{cpi} = -\sum_{(g_i,s_j)\in\mathcal{P}_{cpi}} y_{ij}^{cpi} \log p_{ij}^{cpi} + (1 - y_{ij}^{cpi}) \log(1 - p_{ij}^{cpi}), \tag{9}$$

where $\mathcal{P}_{dti}$ (resp., $\mathcal{P}_{cpi}$) denotes the set of drug-target (resp., compound-protein) pairs in training set and y_{ij}^{dti} (resp., y_{ij}^{cpi}) is the true label of DTI pair (d_i, t_j) (resp., CPI pair (g_i, s_j)). Meanwhile, L2 regularization with a penalty coefficient of 1 is adopted to prevent the model from overfitting. And we adopt a *sigmoid* function to calculate the interaction probability of given pairs.

3.5 Learning Strategy

In multi-task learning, it is necessary to optimize multiple objectives at the same time. A simple way is to directly sum up the losses of multiple tasks, but it cannot adapt to the differences between various tasks. To solve the limitation, in this paper, we introduce a method based on Bayesian uncertainty to alleviate the potential negative risk in multi-task learning [39]. To apply the theory to our sigmoid classifier, we relied on the assumption [40] that $\frac{x}{\lambda^2}(e^{\frac{x}{\lambda^2}} + 1) \approx (e^x + 1)^{\frac{1}{\lambda^2}}$, as it can be simply observed that the equation holds when $\lambda = 1$. Then, considering the sigmoid likelihood and loss functions of the two tasks, the final form of the optimization objective is obtained as follows:

$$\begin{aligned} \mathcal{L}_{total} &= \mathcal{L}(\lambda_1, \lambda_2) \\ &= -\log(P(y_{dti}|f_{dti}(\cdot), \lambda_1) \cdot P(y_{cpi}|f_{cpi}(\cdot), \lambda_2)) \\ &= \frac{1}{\lambda_1^2}\mathcal{L}_{dti} + \frac{1}{\lambda_2^2}\mathcal{L}_{cpi} + \log\lambda_1 + \log\lambda_2, \end{aligned} \tag{10}$$

where λ_1 (resp., λ_2) is trainable parameter of the probability model in DTI (resp., CPI) task, y_{dti} (resp., y_{cpi}) represents the label of DTI (resp., CPI) pair and f_{dti} (resp., f_{cpi}) is the mapping function of DTI (resp., CPI) module.

The pseudocode of jointly optimization procedure for KG-MTL is outlined in Algorithm 1. For the given inputs, P_{cpi} (resp., P_{dti}) and U_{cpi} (resp., U_{dti}) represent the positive and negative samples of CPI (resp., DTI) pairs, and G_{kg} is the large-scale knowledge graph DRKG. At the beginning (Line 1), we use a fixed random seed to initialize all learnable parameters Θ in KG-MTL. Then we split the samples of CPI and DTI pairs into the training, validation and test set, respectively, by a ratio of 8/1/1 (Line 2). For each training iteration, we will sample a subgraph G_{sub} from G_{kg} (Line 4), and then a RGCN module is applied to extract heterogeneous features (i.e., d_o for drug d and t_o for target t) from G_{sub} (Line 7-8). Moreover, in CPI module, we learn compound representation g_o from molecular graph g by GCN model, and we extract the embedding s_o from protein sequence s using linear layer (Line 9). Once

Algorithm 1 Multi task training for KG-MTL

Input: CPI pairs P_{cpi}, U_{cpi}, DTI pairs P_{dti}, U_{dti}, KG G_{kg}, and g, s, d, t represent the compound, protein, drug, target between CPI and DTI pairs, respectively;
Output: $\mathcal{F}(g, s, d, t|\Theta, P_{cpi}, U_{cpi}, P_{dti}, U_{dti}, G_{kg})$;
1: Initialize all parameters;
2: Split training set from P_{cpi}/U_{cpi} and P_{dti}/U_{dti} for CPI and DTI tasks, respectively, by 10-fold cross validation;
3: **for** zero to training epochs **do**
4: Sample subgraph G_{sub} from G_{kg} with negative triples;
5: // CPIs&DTIs prediction tasks with Shared Unit
6: **for** i steps **do**
7: Embed each node of subgraph G_{sub};
8: Extract heterogeneous features d_o, t_o of d and t from $\mathcal{G}_{sub}$ by RGCN module;
9: Represent the compound g and protein s as g_o and s_o by GCN, linear modules and CNN respectively;
10: Obtain the representations of compound g_r and drug d_r from g_o and d_o fused by the Shared Unit module;
11: Predict the potential CPI and DTI using the concatenated vectors $[g_r; s_o], [d_r; t_o]$;
12: Calculate the task-dependent loss function by Eq. (8-9);
13: Calculate the total loss by Eq. (10);
14: Update all parameters of $\mathcal{F}$ by gradient descent;
15: **end for**
16: **end for**

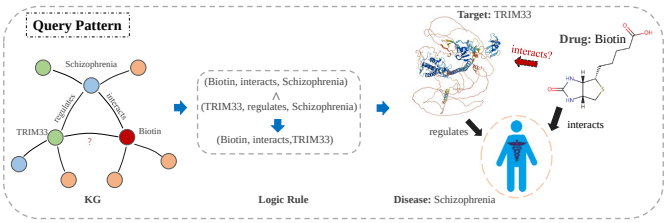


Fig. 2. The query pattern of a potential drug-target interaction in our proposed KG-MTL.

the representations of compound and corresponding drug entity are obtained, we input them into the *Shared Unit* module to output the mixed features g_r, d_r, and then they will be updated in next linear layer and RGCN layer (Line 10). Next, we adopt the stitched vectors (i.e., $[g_r; s_o]$ and $[d_r; t_o]$) to predict CPI and DTI (Line 11). Furthermore, we perform **Learning Strategy** on the loss function with the predicted values (Line 12-13). At last, we update all the trainable parameters (Line 14). The process stops when the model converges.

3.6 Computational Complexity Analysis

The computational complexity of KG-MTL consists of three parts. Specifically, the update of entity embedding for RGCN model in DTI prediction task has the computational complexity of $O(dim^2 NK|R|)$ by Eq. (1), where K is the number of neighbors, N is the number of nodes and $|R|$ represents the number of relations in the knowledge graph. The learning process of molecule graph and protein sequence by GCN and CNN models in CPI prediction task takes $O(n^2 dim)$ and $O(dim^2 F^2)$ respectively, where n is the number of nodes in molecule graph and F represents the size of the kernel in CNN model. The updating of shared features from both two tasks are related to linear and cross operators, so they take $O(dim)$ and $O(dim^2)$ respectively (see *Section Shared Unit*), where dim is the dimension of feature vectors. Therefore, supposing the training stops after i steps, the overall computational complexity is obtained as follows:

$$O(((dimNK|R| + n^2 + dimF^2 + 1 + dim)dim)i),$$

note that $K \ll N$, $n \ll N$ and $|R| \ll N$. The main complexity is matrix multiplication which is also a basic operation in deep graph neural networks. And we can observe that the overall complexity of KG-MTL mainly depends on the feature size dim, the number of relations $|R|$, and the number of nodes in the knowledge graph. For large-scale knowledge graphs and datasets, we speed up the training process in each iteration by using sparse matrix and subgraph sampling.

3.7 Query Pattern of Knowledge Graph

In our paper, the adopted knowledge graph is a comprehensive biological knowledge graph relating to drugs, diseases, proteins, genes, pathways, and expression. It includes 5.9 million edges belonging to 107 types of relationships (e.g., treatment, regulates). In fact, the knowledge graph does not contain the existing drug-target pairs and is only used to extract semantic information from drug and target representations. The learned embedding is adopted to determine whether there is an interaction between drug-target pairs. Fig. 2 shows a query pattern to discover the potential interaction between a drug-target pair. We first obtain some logic rules of a drug-target pair from the adopted knowledge graph (see the middle part of Fig. 2). Then we can observe that the drug *Biotin* (ID: DB00121) has an interaction relation with the disease *Schizophrenia* (Mesh ID: D012559). Meanwhile, the disease *Schizophrenia* can be regulated by target *TRIM33* (Gene ID: 51592). Finally we can further infer that the drug *Biotin* is more likely to interact with target *TRIM33*.

4 Experiments

4.1 Datasets and Settings

We evaluate our proposed KG-MTL[1] by using four datasets: 1) **DrugBank** collects the unique bioinformatics and cheminformatics resources that contain 16,553 drug-target interactions with 5,996 drugs and 3,479 targets [41]. 2) **DrugCentral** contains 9,477 drug-target interactions with 1,427 drugs and 1,106 targets [42]. 3) **human** and 4) **C.elegans** are high quality datasets that integrate various resources [21]. The human dataset contains 2,471 compound-protein interactions with 1,080 compounds and 816 proteins while the C.elegans dataset includes 2,547 compound-protein interactions with 886 compounds and 806 proteins. To provide much structured information on various entities, we adopt a large-scale knowledge graph named DRKG that collects 97,238 entities and 5,874,261 triples belonging to 13 entity-types (e.g., drug, target and disease) and 107 edge-types respectively [32].

4.2 Data processing

The DrugBank and DrugCentral datasets are adopted in DTI task, and we randomly sample from positive samples to generate the same number of the negative DTI pair as positive one since no negative DTI pairs are provided. Subsequently, we take a drug (resp., target) sample with DrugBank ID (resp., protein for Uniprot ID) from training set, and then map the sample ID to the corresponding entity of DRKG to obtain the embedding of drug or target. As to CPI task, the positive and negative samples are unbalanced in human and C.elegans datasets, and thus we adjust the ratio to 1:3 to adapt the prediction model. Besides, we removed these samples whose drug entity can not be found in the knowledge graph to merge the features of multiple tasks from the same molecule samples (i.e., the drug entity and corresponding molecular graph of the compound). After that, we use 10-fold cross-validation and choose two folds as the validation and test sets in each iteration to split the dataset into 8/1/1. To evaluate the performance, we adopt *accuracy* (ACC), *area under the ROC curve* (AUC) and *area under the precision-recall curve* (AUPR) as the metrics.

4.3 Implementation Details

In the training of DTI task, to accelerate the training process and to save GPU (i.e., Graphics Processing Unit) memory,

1. https://github.com/xzenglab/KG-MTL

TABLE 1
Results of DTI task. The first/second row of each method corresponds to the results on DrugCentral and DrugBank respectively.

Methods \ Metrics	ACC	AUC	AUPR
RF	0.832 ± 0.004	0.589 ± 0.005	0.679 ± 0.004
	0.774 ± 0.003	0.636 ± 0.006	0.717 ± 0.004
SVM	0.688 ± 0.001	0.613 ± 0.004	0.590 ± 0.002
	0.624 ± 0.002	0.567 ± 0.001	0.552 ± 0.003
DNN	0.879 ± 0.006	0.941 ± 0.003	0.932 ± 0.007
	0.833 ± 0.005	0.891 ± 0.004	0.891 ± 0.008
TransE	0.853 ± 0.003	0.909 ± 0.012	0.929 ± 0.001
	0.901 ± 0.004	0.924 ± 0.003	0.935 ± 0.011
DistMult	0.912 ± 0.001	0.943 ± 0.002	0.955 ± 0.001
	0.893 ± 0.003	0.927 ± 0.005	0.932 ± 0.005
GCN-KG	0.833 ± 0.004	0.879 ± 0.007	0.893 ± 0.002
	0.894 ± 0.003	0.929 ± 0.004	0.924 ± 0.003
GNN-DTI	0.852 ± 0.004	0.921 ± 0.002	0.913 ± 0.002
	0.761 ± 0.007	0.845 ± 0.006	0.846 ± 0.005
DeepConv-DTI	0.847 ± 0.013	0.903 ± 0.003	0.885 ± 0.007
	0.801 ± 0.009	0.892 ± 0.007	0.893 ± 0.004
DeepDTI	0.866 ± 0.007	0.813 ± 0.001	0.846 ± 0.013
	0.636 ± 0.010	0.729 ± 0.002	0.778 ± 0.010
TriModel	0.812 ± 0.003	0.883 ± 0.004	0.871 ± 0.001
	0.873 ± 0.001	0.934 ± 0.005	0.941 ± 0.001
NeoDTI	0.882 ± 0.007	0.923 ± 0.001	0.895 ± 0.016
	0.891 ± 0.002	0.951 ± 0.005	0.917 ± 0.003
KG-MTL	**0.964** $\pm 0.001 \uparrow$	**0.980** $\pm 0.001 \uparrow$	**0.982** $\pm 0.001 \uparrow$
	0.940 $\pm 0.003 \uparrow$	**0.959** $\pm 0.004 \uparrow$	**0.959** $\pm 0.003 \uparrow$
KG-MTL-S_{dti}	0.905 ± 0.006	0.946 ± 0.004	0.946 ± 0.004
	0.878 ± 0.004	0.929 ± 0.002	0.926 ± 0.003
KG-MTL-L	0.940 ± 0.004	0.969 ± 0.005	0.967 ± 0.007
	0.931 ± 0.004	0.946 ± 0.003	0.943 ± 0.004
KG-MTL-C	0.934 ± 0.005	0.965 ± 0.005	0.964 ± 0.006
	0.928 ± 0.004	0.946 ± 0.003	0.949 ± 0.005

TABLE 2
Performance of CPI task. The first/second row of each method corresponds to the results on human and C.elegans respectively.

Methods \ Metrics	ACC	AUC	AUPR
RF	0.883 ± 0.002	0.422 ± 0.002	0.433 ± 0.007
	0.912 ± 0.003	0.417 ± 0.009	0.431 ± 0.005
SVM	0.811 ± 0.001	0.515 ± 0.008	0.412 ± 0.018
	0.839 ± 0.004	0.417 ± 0.003	0.372 ± 0.003
DNN	0.877 ± 0.002	0.910 ± 0.002	0.835 ± 0.005
	0.911 ± 0.007	0.962 ± 0.002	0.922 ± 0.005
TransE	0.893 ± 0.002	0.936 ± 0.008	0.879 ± 0.001
	0.854 ± 0.001	0.927 ± 0.004	0.930 ± 0.009
DistMult	0.881 ± 0.004	0.937 ± 0.005	0.878 ± 0.006
	0.901 ± 0.012	0.946 ± 0.006	0.926 ± 0.003
GCN-KG	0.843 ± 0.002	0.891 ± 0.005	0.889 ± 0.013
	0.904 ± 0.001	0.932 ± 0.003	0.912 ± 0.003
GNN-CPI	0.871 ± 0.013	0.916 ± 0.002	0.856 ± 0.009
	0.843 ± 0.001	0.781 ± 0.008	0.713 ± 0.004
DeepConv-CPI	0.866 ± 0.003	0.902 ± 0.008	0.844 ± 0.002
	0.856 ± 0.001	0.934 ± 0.007	0.825 ± 0.003
GraphCPI	0.747 ± 0.026	0.899 ± 0.001	0.781 ± 0.013
	0.828 ± 0.026	0.943 ± 0.001	0.855 ± 0.002
NeoDTI	0.892 ± 0.008	0.881 ± 0.045	0.795 ± 0.083
	0.877 ± 0.007	0.910 ± 0.006	0.763 ± 0.021
KG-MTL	**0.907** $\pm 0.005 \uparrow$	**0.949** $\pm 0.002 \uparrow$	**0.899** $\pm 0.005 \uparrow$
	0.928 $\pm 0.003 \uparrow$	**0.969** $\pm 0.002 \uparrow$	**0.933** $\pm 0.005 \uparrow$
KG-MTL-S_{cpi}	0.876 ± 0.008	0.920 ± 0.004	0.851 ± 0.008
	0.905 ± 0.004	0.931 ± 0.002	0.913 ± 0.004
KG-MTL-L	0.886 ± 0.004	0.921 ± 0.005	0.846 ± 0.007
	0.904 ± 0.004	0.955 ± 0.003	0.918 ± 0.004
KG-MTL-C	0.891 ± 0.005	0.923 ± 0.005	0.849 ± 0.006
	0.907 ± 0.005	0.957 ± 0.003	0.921 ± 0.005

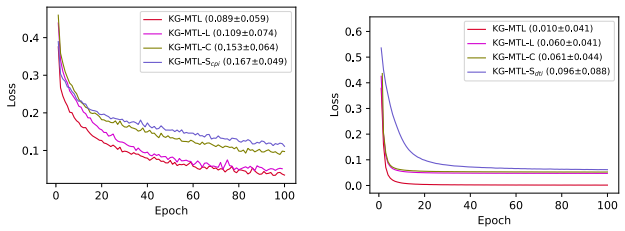
we adopt neighbor sampling to generate a subgraph of 40,000 edges from the knowledge graph DRKG [43]. Then we construct a 3-layer RGCN model with a hidden size of 128 and the dimension of entity embedding is set to 128 as well. And the initialization of entity embedding and relation weights are derived from a normalized distribution $U[-\frac{6}{\sqrt{dim}}, \frac{6}{\sqrt{dim}}]$, where dim is the dimension of the embedding [44]. As to CPI task, we use the GCN layer for molecular graph and output a global embedding with a dimension of 128. For two tasks, we adopt 3 fully-connected layers with 128 hidden units and a sigmoid layer to output the interaction probability for given pair. To optimize all trainable parameters, we use Adam optimizer [45] with a learning rate of 0.001 and save the best model based on the AUC metric of the validation set. And we set the number of *Shared Unit* layer to 1. The batch size and epoch are 32 and 100 respectively. In this paper, we adopt a 10-fold cross-validation to evaluate the performance of KG-MTL, and the mean and standard deviation of all metrics are reported.

4.4 Baselines

To validate the performance of KG-MTL, we compare it with the following state-of-the-art baselines:

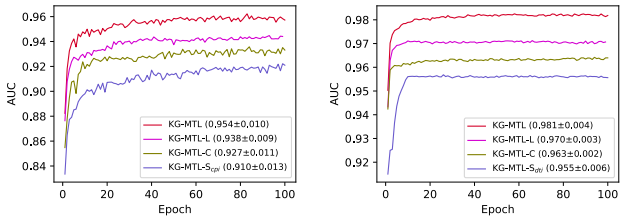
- **RF** (Random Forest), **SVM** (Support Vector Machine) and **DNN** (Deep Neural Network) applied the molecular fingerprints (ECFP) of drug or compound and the PSC features of protein descriptors, and DNN used a three layer DNN with hidden size of 1,024 [46].
- **DeepDTI** [47] applied a neural network based on restricted boltzman machine using ECFP and PSC features for DTI prediction.
- **DeepConv-DTI** [48] adopted CNN to extract the local features of protein sequences and used fully connected layer to encode the molecular fingerprints of drugs for DTI prediction. Here we implemented DeepConv-DTI to predict CPI and termed it as **DeepConv-CPI**.
- **GNN-CPI** [15] applied GNN to encode molecular graph of compounds and adopted CNN to obtain the chemical features of proteins for CPI task. And we implemented GNN-CPI for DTI prediction, denoted by **GNN-DTI**.
- **GraphCPI** [36] extracted the molecular structures of compounds and the chemical contexts of protein sequences by developing the GCN and CNN modules.
- **NeoDTI** [19] constructed a heterogeneous networks to learn latent representations of drugs and targets. We set the dimension of the edge-type projection matrices as 512, and the learning rate to 0.001.
- **TriModel** [25] is an end-to-end model using KG embedding approach for DTI task. Following the original work, we adopt AMSGrad optimizer with a learning rate of 0.01 to optimize the training loss.
- **TransE** [49] and **DistMult** [50] are knowledge graph embedding models that learn the representation of entities, which can be directly used in DTI and CPI tasks. All hyperparameters (e.g., batch size and learning rate) of the two models are kept the same as ours.
- **GCN-KG** adopts the **GCN** [35] model to learn the representations of entities on homogeneous KG in downstream tasks (i.e., DTI and CPI tasks), All hyperparameters are kept the same as ours.

All baselines are based on the public code where we kept the settings of models the same as reported in the original



(a) loss curves of models on human dataset for CPI task. (b) loss curves of DTI task on DrugCentral dataset.

Fig. 3. The loss curves of different variants of KG-MTL on DrugCentral&human dataset. And the loss is calculated on validation set.



(a) AUC curves of models on human dataset for CPI task. (b) AUC curves of DTI task on DrugCentral dataset.

Fig. 4. The AUC curves of various variants of KG-MTL on DrugCentral&human dataset. And the AUC is calculated on test set.

papers. Following [47] and [46], we implemented RF, SVM and DNN models for DTI and CPI prediction respectively.

4.5 DTI Prediction Results

As shown in Table 1, we observe that KG-MTL outperforms all other baselines. Specifically, KG-MTL improves the ACC, AUC, and AUPR by at least 8.2%, 3.9% and 5% respectively on the DrugCentral dataset, and 4.9%, 0.8% and 1.8% respectively on the DrugBank dataset. The improvement indicates that (i) compared with the methods (e.g., DeepConvDTI) that only learn representations of drug and protein sequence, our method can preserve more useful information on various drug-like compounds by the CPI module; and (ii) compared with KG-based models (i.e., TriModel, TransE, DistMult and GCN-KG) that learn node embedding directly, the *Shared Unit* also helps the model to jointly learn the molecular structures and the semantic relations of the drug in DRKG, thus improving the performance of DTI task.

4.6 CPI Prediction Results

The comparison results on the CPI task are listed in Table 2. The results illustrate that KG-MTL outperforms all the baselines across human and C.elegans datasets. More specifically, KG-MTL achieves at least 2.6% on AUC, 1.1% on AUPR higher performance than other methods on the C.elegans dataset. Meanwhile, KG-MTL achieves the best AUC score of 94.9% with at least 3.3% absolute gain compared to GNN-CPI (the second-best method) in the human dataset. The improvement is attributed to the abundant information brought by the DTI module that can extract the semantic relations of drug entities from the knowledge graph, while other methods (e.g., GNN-CPI and NeoDTI) only learn embeddings from molecular structure of compound or the topology of the drug-related network. Meanwhile, compared with the KG-based models (i.e., TransE, DistMult and GCN-KG) that directly adopt knowledge graph information and ignore the molecular structure, KG-MTL has a better performance by fusing KG information and drug structure through the *Shared Unit*.

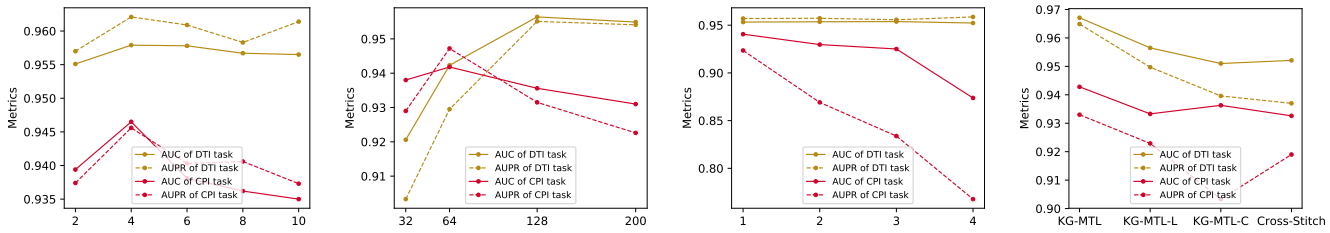
4.7 Ablation Experiments

To investigate how the different operations of *Shared Unit* and learning strategies improve the performance of the proposed model, we conduct the ablation study on the following variants of KG-MTL:

- **KG-MTL-S** is the variant of KG-MTL that removes both the *Shared Unit* and learning strategies. So we can adopt KG-MTL-S$_{dti}$ (resp., KG-MTL-S$_{cpi}$) represents the single DTI task (resp., CPI task)
- **KG-MTL-L** removes cross operation of *Shared Unit* and simply retains the linear operation only.
- **KG-MTL-C** removes linear operation of *Shared Unit* and retains the cross operation.

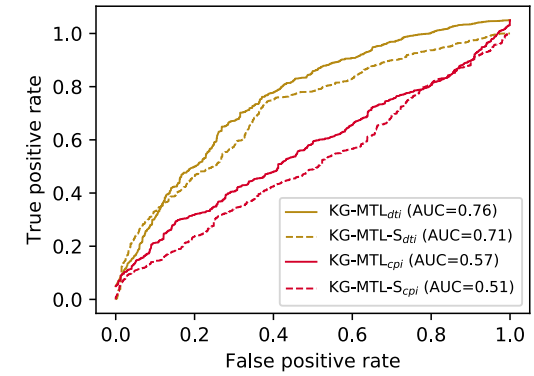
The ablation experiments results on both tasks are shown in Table 1 and Table 2. The results prove that the *Shared Unit* including linear and cross operation, and learning strategy are all effective for both two tasks. Among all the variants, KG-MTL-S has the most significant performance gaps compared with KG-MTL, which indicates that *Shared Unit* contributes the most to help the model to jointly capture the drug features extracted from molecule graph and knowledge graph that improves the prediction performance. Moreover, our proposed method provides better performance than KG-MTL-L and KG-MTL-C in all datasets, which proves that the *Shared Unit* with complete settings is beneficial to improving the prediction performance.

To further validate the effectiveness and stability of *Shared Unit* on DTI and CPI tasks. Fig. 3 and Fig. 4 show the loss and AUC curves of different variants of KG-MTL, respectively. As shown in Fig. 3, we overall observe that KG-MTL provides better robustness and stability with faster convergence and lower loss with comparison to KG-MTL-L and KG-MTL-C. And the loss curve of KG-MTL-S with slower convergence gives limited performance gain on both tasks. We believe that such significant improvements can be attributed to the shared features of drugs or compounds learned from *Shared Unit*, thereby having a positive influence on the performance of KG-MTL. Furthermore, KG-MTL-L achieves the same performance as our KG-MTL in terms of convergence with slower speed, which can further prove that the cross operation of *Shared Unit* is clearly helpful to the speed of model convergence. The reason might be that the cross operation can make the fused features smoother and sparser, which speeds up the training process of KG-MTL. Meanwhile, compared with KG-MTL, we also notice that KG-MTL-C shows the comparative performance on convergence, which indicates that the linear transformation in *Shared Unit* can lower the convergence level of our model. Similarly, as shown in Fig. 4, KG-MTL achieves at least up to 1.6% and 1.1% improvements on the two datasets with comparison to the best baseline method. These findings

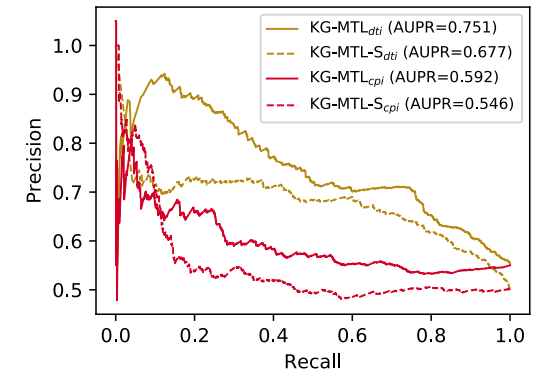


(a) Negative sample size. (b) Embedding dimension. (c) Number of shared unit. (d) Settings of shared unit.

Fig. 5. Results of KG-MTL with varying settings of r, dim, N and *Shared Unit* on DrugCentral and C.elegans datasets respectively.



(a) ROC curves of KG-MTL and KG-MTL-S on BindingDB.



(b) P-R curves of KG-MTL and KG-MTL-S on BindingDB.

Fig. 6. The performance of KG-MTL and KG-MTL-S (including KG-MTL-S$_{dti}$ and KG-MTL-S$_{cpi}$) tested on BindingDB dataset.

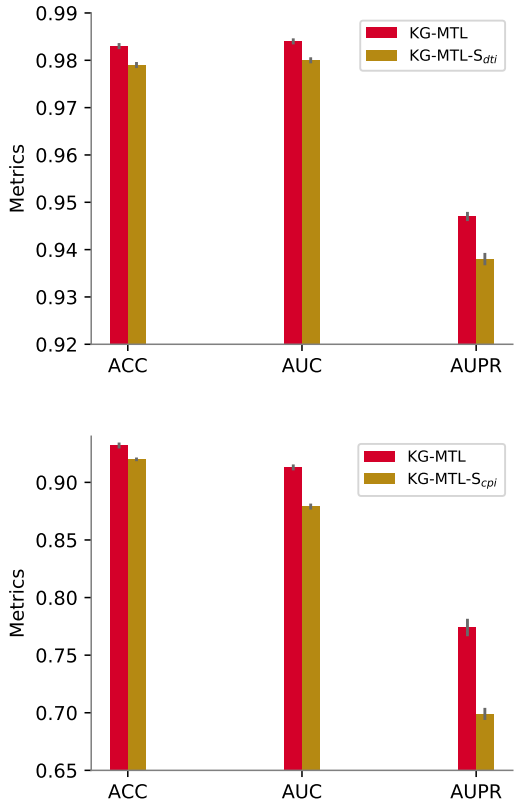


Fig. 7. The results of KG-MTL and KG-MTL-S validated on unbalanced DrugCentral (top) and human (bottom) datasets.

further validate the stability and effectiveness of our proposed *Shared Unit* in KG-MTL. In addition, we further study the changes in the embedding space of drug representations learned from KG-MTL and Cross-Stitch. More details can be found in *Appendix A.1*.

4.8 Parameter Sensitivity Analysis

In this experiment, we test the impact of the major hyper-parameters of KG-MTL.

Impact of Negative Sample Size in KG. As shown in Fig. 5(a), we vary different negative sample size r and observe that the optimal solution can be reached when r=4. This is because KG-MTL can learn more useful information with enough negative samples. However, as the proportion of negative samples increases, some potential positive triples may be treated as negative samples that result in negative effect for the performance of KG-MTL.

Impact of Dimension of Entity Embedding. We investigate the influence of dimension of entity embedding dim by varying it from 32 to 300. Fig. 5(b) illustrates that our method achieves the best AUC when dim=128 in the DTI task, while the best result can be obtained when dim=64 in CPI task. The reason could be that the embedding with relatively larger *dim* can represent much information from large-scale KG in the DTI task, while the molecular features with higher *dim* will lead to information redundancy in CPI task.

Impact of Various Number and Settings of Shared Unit. As shown in Fig. 5(c), we investigate the effect of the number

TABLE 3
Candidate drugs and possible interaction proteins related to COVID-19. Note that NA represents no evidence to identify the fact.

DrugBank ID	Drug Name	Prediction Score	Target	Active protein	Evidence
DB00619	Imatinib	1.0	TNF-α	CPE,AlphaLISA, CoV-PPE,MERS-PPE	NCT04338698
DB12612	Ozanimod	0.99	TNF-α	ACE2,AlphaLISA,CoV-PPE,MERS-PPE	NCT04405102
DB00198	Oseltamivir	0.99	TNF-α	NA	NCT04338698
DB09552	Tonzonium	0.99	TNF-α	AlphaLISA,3CL enzymatic activity,MERS-PPE	NA
DB09220	Nicorandil	0.98	TNF-α	NA	PMC7436472
DB01268	Sunitinib	0.99	IL-6	AlphaLISA,MERS-PPE	PMC7550610
DB00811	Ribavirin	0.99	IL-6	NA	NCT04494399
DB01143	Amifostone	0.98	IL-6	3CL enzymatic activity,CoV-PPE	PMC3661204
DB09079	Nintedanib	0.98	IL-6	ACE2,CoV-PPE,CPE,AlphaLISA	PMC7969149
DB00284	Acarbose	0.98	IL-6	NA	PMC3832586

of *Shared Unit* N by varying it from 1 to 4 (Recall the *Shared Unit* in Section 3.2). We find that KG-MTL achieves worse performance in CPI task as N increases, while it obtains the stable AUC score in DTI task. This implies that the shared features in lower layer will be more beneficial to improving the performance of model. Meanwhile, Fig. 5(d) shows the influence of *Shared Unit* with different settings. We observe that the *Shared Unit* with both linear and cross operations achieves better performance than other operations (e.g., *Cross-Stitch* [26]). This proves that KG-MTL can effectively leverage the semantic relations and molecular structures of drugs by using *Shared Unit* in high-order feature space, while *Cross-Stitch*(resp., KG-MTL-L) directly adopted four trainable parameters to share task-independent features.

.

4.9 Generalization of KG-MTL

To validate the effectiveness and generalization of our proposed *Shared Unit* in multi tasks, we adopt an external dataset BindingDB [51] to evaluate the performance of KG-MTL and the result is shown in Fig. 6. Following earlier work [46], we collect the positive DTI/CPI pairs that satisfy $k_d < 30$ units and sample the same number of negative DTI/CPI pairs as the positive samples for external validation set. We plot the ROC curves of KG-MTL and KG-MTL-S in Fig. 6(a). We find that KG-MTL achieves superior performance over the single-task models (i.e., KG-MTL-S$_{dti}$ and KG-MTL-S$_{cpi}$). Moreover, we also plot the precision-recall curves in Fig. 6(b). Specifically, KG-MTL achieves higher AUPR results of 7.4% for DTI task and 4.6% for CPI tasks, respectively, which indicates that our proposed KG-MTL is more beneficial to predict the unknown molecular interactions than single-task models. The results demonstrate that KG-MTL can learn more generalized features of drugs via the *Shared Unit* between multiple tasks.

However, the number of known DTI or CPI pairs is much smaller than the unknown one, which leads to a serious imbalance in two datasets. To mimic the situation, we perform a cross-validation test that the negative samples in the test set contain nine times more than the positive ones [18]. Thus, the positive samples (i.e., known DTIs/CPIs) occupy only 10% of the whole dataset in the setting of unbalanced datasets. In previous works [52], [53], as the *area under the roc curve* (AUC) may be an over-optimistic metric to evaluate the performance of model in highly imbalanced dataset, here we add the AUPR metric to give a better evaluation in this scenario. As shown in Fig. 7, we observe that the AUPR results of KG-MTL and KG-MTL-S are declined on the unbalanced dataset with comparison to their performance on DrugCentral and human datasets, but we also notice that KG-MTL still achieves a higher AUPR score than the single-task model (i.e., KG-MTL-S). This further implies that multi-task learning of our KG-MTL results in a quick adaptation to predict unknown DTIs/CPIs on the sparse datasets and a much more significant improvement in generalization. Thus, KG-MTL indeed narrows the gap by learning how to make adaptations on the unbalanced dataset. Meanwhile, we conduct a comparison experiment on two new test data with a reasonable split. More details are shown in *Appendix A.2* and *A.3*.

4.10 Case Study: COVID-19

Lastly, we further present a case study to show the potential predictive ability of KG-MTL. Table 3 shows the top 10 drugs that predicted by our model are selected as the candidate agents binding to TNF-α and IL-6. We observe that nine drugs can be confirmed. For example, Nicorandil (ID:PMC7436472) and Acarbose (ID:PMC3832586) have been reported by PubMed. Meanwhile, Imatinib and Ribavirin are in clinical trials, and the evidence can be checked by their NCT number. In addition, we use DrugCentral REDIAL 2020 [54] toolkit to evaluate the drug activities to the Sars-CoV-2 related targets. And we also find that these drugs have high ACE2[2] enzymatic activity or 3CL[3] enzymatic activity for COVID-19 (e.g., Amifostone, Ozanimod), which further proves the superiority of the KG-MTL. The knowledge graph enhanced multi-task learning framework is a promising tool for predicting the potential drug-target interactions.

5 Conclusion

Molecular interaction prediction (e.g., DTI prediction and CPI prediction) between targets plays a key role in many applications, including pharmacology and clinical application. In this paper, we focus on molecular interaction prediction that demands the model to capture the features of drug and

2. https://opendata.ncats.nih.gov/covid19/assay?aid=6
3. https://opendata.ncats.nih.gov/covid19/assay?aid=9

the interactions related to targets. However, previous works represent drug features with insufficient information and ignore semantic information in knowledge graph. To address this limitation, we propose a novel framework named KG-MTL that develops a novel shared unit in the view of multi-task learning, to capture the information from both molecular graph of compounds and semantic relations of drug entities of knowledge graph respectively. Experimental results on real-world datasets show that KG-MTL could improve the performance on the drug-target interaction prediction and compound-protein interaction prediction tasks.

ACKNOWLEDGMENTS

The authors would like to thank all the reviewers for their insightful and valuable suggestions, which significantly improve the quality of this paper. The work was supported in part by National Natural Science Foundation of China [62122025, 61872309, 61972138, 62102140], the Hunan Provincial Natural Science Foundation of China [2020JJ4215, 2021JJ10020, 2022JJ40451], and the NSF under grants [III-1763325, III-1909323, III-2106758, SaTC-1930941].

REFERENCES

[1] M. Lukačišin and T. Bollenbach, "Emergent gene expression responses to drug combinations predict higher-order drug interactions," *Cell Systems*, vol. 9, no. 5, pp. 423–433, 2019.

[2] M. Bredel and E. Jacoby, "Chemogenomics: an emerging strategy for rapid target and drug discovery," *Nature Reviews Genetics*, vol. 5, no. 4, pp. 262–275, 2004.

[3] Y.-s. Lee, A. Krishnan, R. Oughtred, J. Rust, C. S. Chang, J. Ryu, V. N. Kristensen, K. Dolinski, C. L. Theesfeld, and O. G. Troyanskaya, "A computational framework for genome-wide characterization of the human disease landscape," *Cell systems*, vol. 8, no. 2, pp. 152–162, 2019.

[4] C. Zang and F. Wang, "Moflow: an invertible flow model for generating molecular graphs," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 617–626.

[5] Z. Hao, C. Lu, Z. Huang, H. Wang, Z. Hu, Q. Liu, E. Chen, and C. Lee, "Asgn: An active semi-supervised graph neural network for molecular property prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 731–752.

[6] M. Sun, F. Wang, O. Elemento, and J. Zhou, "Structure-based drug-drug interaction detection via expressive graph convolutional networks and deep sets (student abstract)," in *Association for The Advancement of Artificial Intelligence*, 2020, pp. 13 927–13 928.

[7] T. Fu, C. Xiao, L. Glass, and J. Sun, "Moler: Incorporate molecule-level reward to enhance deep generative model for molecule optimization," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[8] C. Xiao, P. Zhang, W. A. Chaowalitwongse, J. Hu, and F. Wang, "Adverse drug reaction prediction with symbolic latent dirichlet allocation," in *Association for The Advancement of Artificial Intelligence*, 2017, pp. 1590–1596.

[9] S. Dey, P. Zhang, D. Sow, and K. Ng, "Perdrep: Personalized drug effectiveness prediction from longitudinal observational data," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1258–1268.

[10] I. Shaked, M. A. Oberhardt, N. Atias, R. Sharan, and E. Ruppin, "Metabolic network prediction of drug side effects," *Cell systems*, vol. 2, no. 3, pp. 209–213, 2016.

[11] F. Ma, C. Meng, H. Xiao, Q. Li, J. Gao, L. Su, and A. Zhang, "Unsupervised discovery of drug side-effects from heterogeneous data sources," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 967–976.

[12] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug-target interactions using bipartite local models," *Bioinformatics*, vol. 25, no. 18, pp. 2397–2403, 2009.

[13] T. van Laarhoven, S. B. Nabuurs, and E. Marchiori, "Gaussian interaction profile kernels for predicting drug-target interaction," *Bioinformatics*, vol. 27, no. 21, pp. 3036–3043, 2011.

[14] K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, and P. Zhang, "Interpretable drug target prediction using deep neural representation," in *International Joint Conference on Artificial Intelligence*, 2018, pp. 3371–3377.

[15] M. Tsubaki, K. Tomii, and J. Sese, "Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences," *Bioinformatics*, vol. 35, no. 2, pp. 309–318, 2019.

[16] H. Chen and J. Li, "Learning data-driven drug-target-disease interaction via neural tensor network," in *International Joint Conference on Artificial Intelligence*, 2020, pp. 3452–3458.

[17] W. Zhao, J. Zhu, M. Yang, D. Xiao, G. P. C. Fung, and X. Chen, "A semi-supervised network embedding model for protein complexes detection," in *Association for The Advancement of Artificial Intelligence*, 2018, pp. 8185–8186.

[18] Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen, and J. Zeng, "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information," *Nature communications*, vol. 8, no. 1, pp. 1–13, 2017.

[19] F. Wan, L. Hong, A. Xiao, T. Jiang, and J. Zeng, "Neodti: neural integration of neighbor information from a heterogeneous network for discovering new drug-target interactions," *Bioinformatics*, vol. 35, no. 1, pp. 104–111, 2019.

[20] M. Tognetti, A. Gabor, M. Yang, V. Cappelletti, J. Windhager, O. M. Rueda, K. Charmpi, E. Esmaeilishirazifard, A. Bruna, N. de Souza *et al.*, "Deciphering the signaling network of breast cancer improves drug sensitivity prediction," *Cell Systems*, vol. 12, no. 5, pp. 401–418, 2021.

[21] H. Liu, J. Sun, J. Guan, J. Zheng, and S. Zhou, "Improving compound-protein interaction prediction by building up highly credible negative samples," *Bioinformatics*, vol. 31, no. 12, pp. i221–i229, 2015.

[22] K. Huang, C. Xiao, T. Hoang, L. Glass, and J. Sun, "Caster: Predicting drug interactions with chemical substructure representation," in *Association for The Advancement of Artificial Intelligence*, 2020, pp. 702–709.

[23] X. Lin, Z. Quan, Z.-J. Wang, T. Ma, and X. Zeng, "KGNN: knowledge graph neural network for drug-drug interaction prediction," in *International Joint Conference on Artificial Intelligence*, C. Bessiere, Ed., 2020, pp. 2739–2745.

[24] J. Shang, C. Xiao, T. Ma, H. Li, and J. Sun, "Gamenet: Graph augmented memory networks for recommending medication combination," in *Association for The Advancement of Artificial Intelligence*, vol. 33, 2019, pp. 1126–1133.

[25] S. K. Mohamed, V. Nováček, and A. Nounu, "Discovering protein drug targets using knowledge graph embeddings," *Bioinformatics*, vol. 36, no. 2, pp. 603–610, 2020.

[26] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[27] J. Xu, J. Zhou, P.-N. Tan, X. Liu, and L. Luo, "Spatio-temporal multi-task learning via tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2764–2775, 2019.

[28] H. Xiao, Y. Chen, and X. Shi, "Knowledge graph embedding based on multi-view clustering framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 585–596, 2019.

[29] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, "Multi-channel deep networks for block-based image compressive sensing," *IEEE Transactions on Multimedia*, vol. 23, pp. 2627–2640, 2021.

[30] S. Li, F. Wan, H. Shu, T. Jiang, D. Zhao, and J. Zeng, "Monn: a multi-objective neural network for predicting compound-protein interactions and affinities," *Cell Systems*, vol. 10, no. 4, pp. 308–322, 2020.

[31] G. Landrum, "Rdkit: Open-source cheminformatics," 2006.

[32] V. N. Ioannidis, X. Song, S. Manchanda, M. Li, X. Pan, D. Zheng, X. Ning, X. Zeng, and G. Karypis, "Drkg - drug repurposing knowledge graph for covid-19," https://github.com/gnn4dr/DRKG/, 2020.

[33] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *International Conference on Learning Representations*, 2018, pp. 1–15.

[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, 2018, pp. 593–607.
[35] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ArXiv*, vol. abs/1609.02907, 2017.
[36] Z. Quan, Y. Guo, X. Lin, Z.-J. Wang, and X. Zeng, "Graphcpi: Graph neural representation learning for compound-protein interaction," in *IEEE International Conference on Bioinformatics and Biomedicine*, 2019, pp. 717–722.
[37] B. Dzmitry, C. Kyunghyun, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *The International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2014.
[38] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.
[39] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482–7491.
[40] H. Li, Y. Wang, Z. Lyu, and J. Shi, "Multi-task learning for recommendation over heterogeneous information network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 789–802, 2020.
[41] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda *et al.*, "Drugbank 5.0: a major update to the drugbank database for 2018," *Nucleic Acids Research*, vol. 46, p. D1074–D1082, 2018.
[42] O. Ursu, J. Holmes, J. Knockel, C. G. Bologa, J. J. Yang, S. L. Mathias, S. J. Nelson, and T. I. Oprea, "DrugCentral: online drug compendium," *Nucleic Acids Research*, vol. 45, no. D1, pp. D932–D939, 2016.
[43] F. Costa and K. De Grave, "Fast neighborhood subgraph pairwise distance kernel," in *International Conference on Machine Learning*, 2010, pp. 255–262.
[44] Y. Zhao, A. Zhang, R. Xie, K. Liu, and X. Wang, "Connecting embeddings for knowledge graph entity typing," in *Association for Computational Linguistics*, 2020, pp. 6419–6428.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[46] K. Huang, C. Xiao, L. Glass, and J. Sun, "Moltrans: Molecular interaction transformer for drug target interaction prediction," *Bioinformatics*, pp. 1–7, 2020.
[47] M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, and H. Lu, "Deep learning-based drug-target interaction prediction," *Journal of proteome research*, vol. 16, no. 4, pp. 1401–1409, 2017.
[48] I. Lee, J. Keum, and H. Nam, "Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences," *PLoS Computational Biology*, vol. 15, no. 6, p. e1007129, 2019.
[49] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.
[50] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.
[51] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, "Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities," *Nucleic acids research*, vol. 35, no. suppl_1, pp. D198–D201, 2007.
[52] T. van Laarhoven, S. B. Nabuurs, and E. Marchiori, "Gaussian interaction profile kernels for predicting drug–target interaction," *Bioinformatics*, vol. 27, no. 21, pp. 3036–3043, 2011.
[53] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *International conference on Machine learning*, 2006, pp. 233–240.
[54] G. Bocci, S. Verma, M. M. Hassan, J. Holmes, J. J. Yang, S. Sirimulla, T. I. Oprea *et al.*, "A machine learning platform to estimate anti-sars-cov-2 activities," *Nature Machine Intelligence*, pp. 1–9, 2021.

Tengfei Ma received the BS degree from School of Software and Applied Science and Technology, Zhengzhou University, China, in 2019. He is a postgraduate student at the School of Information Science and Engineering, Hunan University. His research interests include drug discovery, graph neural network, knowledge graph representation learning. He has published several research works in these fields including IJCAI, J. Proteome Res, Bioinformatics, etc.

Xuan Lin is currently a lecturer at the College of Computer Science, Xiangtan University, Xiangtan, China. Before joining Xiangtan University, he received the PhD degree in computer science from Hunan University, Changsha, China, in 2021. He was visiting scholar in University of Illinois at Chicago, from 2019 to 2020. His main research interests include machine learning, graph neural networks and bioinformatics. He has published several research papers in these fields including IJCAI, AAAI, ECAI, BIBM, Briefings in Bioinformatics, etc.

Bosheng Song received the Ph.D. degree in control science and engineering from Huazhong University of Science and Technology, Wuhan, China, in 2015. He spent 18 months working with the Research Group on Natural Computing, University of Seville, Seville, Spain, from 2013 to 2015. He was a Postdoctoral Researcher with the School of Automation, Huazhong University of Science and Technology, from 2016 to 2019. He is currently an Associate Professor with the College of Information Science and Engineering, Hunan University, Changsha, China. His current research interests include membrane computing and bioinformatics.

Philip S Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. His research interest is on big data, including data mining, data stream, database and privacy. He has published more than 1,000 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. He received the ICDM 2013 10-year Highest-Impact Paper Award, and the EDBT Test of Time Award (2014). He is a Fellow of the ACM and the IEEE.

Xiangxiang Zeng is an Yuelu distinguished Professor with the College of Information Science and Engineering, Hunan University, Changsha, China. Before joining Hunan University in 2019, he was with Department of Computer Science in Xiamen University. He received his Ph.D. degree in system engineering from Huazhong University of Science and Technology, China, in 2011. He was visiting scholar in Indiana University, The Chinese University of Hongkong, Oklahoma State University, etc. His main research interests include computational intelligence, graph neural networks and bioinformatics. He is a senior member of the IEEE.