Improving Multi-Task GNNs for Molecular Property Prediction via Missing Label Imputation

Journal:	*Machine Intelligence Research*
Manuscript ID	MIR-2023-02-016.R1
Manuscript Type:	Research Article
Date Submitted by the Author:	17-Mar-2023
Complete List of Authors:	Hu, Fenyu Chen, Dingshuo; Chinese Academy of Sciences Institute of Automation Liu, Qiang; Chinese Academy of Sciences Institute of Automation Wu, Shu
Keywords:	machine learning, pattern recognition, bioinformatics, applied chemistry
Specialty/Area of Expertise:	Semi-Supervised Learning < 3. Pattern Recognition & Machine Learning, Mining Graphs, Semi Structured Data, Complex Data < 7. Knowledge Discovery & Data Mining, AI in Medicine and Healthcare < 8. Applications

Improving Multi-Task GNNs for Molecular Property Prediction via Missing Label Imputation

Fenyu Hu[1,2], Dingshuo Chen[1,2], Qiang Liu[1,2] and Shu Wu[1,2]

[1]University of Chinese Academy of Sciences, Beijing 100190, China.

[2]Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Abstract

The prediction of molecular properties is a fundamental task in the field of drug discovery. Recently, Graph Neural Networks (GNNs) have been gaining prominence in this area. Since a molecule tends to have multiple correlated properties, there is a great need to develop the multi-task learning ability of GNNs. However, limited by expensive and time-consuming human annotations, collecting complete labels for each task is difficult. As a result, most existing benchmarks involve a lot of missing labels in training data, and the performance of GNNs is impaired for lacking enough supervision information. To overcome this obstacle, we propose to improve multi-task molecular property prediction via missing label imputation. Specifically, a bipartite graph is firstly introduced to model the molecule-task co-occurrence relationships. Then, the imputation of missing labels is transformed into predicting missing edges on this bipartite graph. To predict the missing edges, a graph neural network is devised, which can learn the complex molecule-task co-occurrence relationships. After that, we select reliable pseudo-labels according to the uncertainty of the prediction results. Boosting with enough and reliable supervision information, our approach achieves the state-of-the-art performance on a variety of real-world datasets.

Keywords: Graph classification, imbalance learning, prediction bias, mixture of experts, multi-view representations.

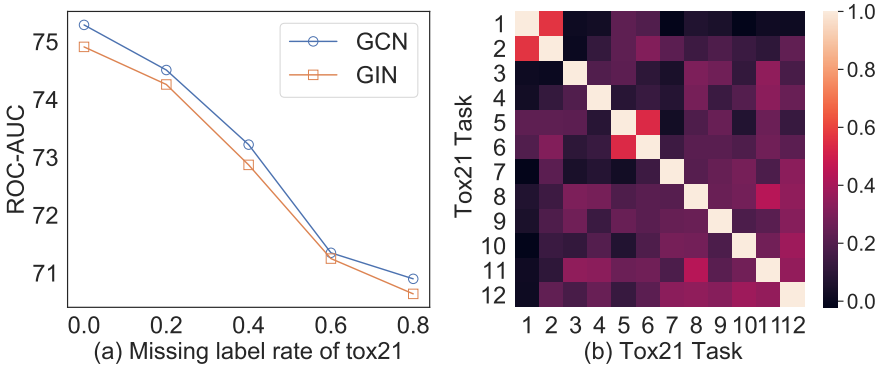


Fig. 1: (a) Test ROC-AUC on the Tox21 dataset under different missing lable rates. (b) Correlation heatmap of tasks on the training set of Tox21, where each element represents the Pearson correlation coefficient.

1 Introduction

Molecular property prediction has a significant impact on drug discovery. Over the past few decades, both industry and academia [1, 2] have paid close attention to machine learning methods to improve the prediction performance. Usually, the multi-task learning ability of the models is required [3, 4] since a molecule tends to have multiple correlated properties. Herein, each task refers to predicting the molecular property towards a specific biological target. Previous studies [3, 5] have shown that multi-task learning can significantly boost the model performance compared with single-task methods. Therefore, it is of high application value to design a multi-task model for molecular property prediction.

Recently, Graph Neural Networks (GNNs) have emerged as a successful method for molecular property prediction. Specifically, a molecule is represented as a graph, where nodes denote atoms, and edges represent chemical bonds. Correspondingly, the prediction of molecular property can be regarded as a graph classification problem. Existing GNNs usually concentrate on designing better model architectures, such as message passing schemes [6], pooling functions [7], expressive power [8] and normalization strategies [9].

Despite the prosperous development of GNNs on molecular property prediction, the issue of missing labels in the multi-task setting remains rarely explored in existing literature. In real-life applications, missing labels are quite common and seriously impair the performances of current approaches. Specifically, there are usually thousands of molecular graphs along with hundreds of properties, and thus collecting complete labels is costly and time-consuming. We perform a statistical analysis of the datasets in MoleculeNet [10], which is a well-known benchmark for molecular property prediction. We find that the missing label rate reaches up to 84% and 71% in the Muv and Toxcast dataset,

respectively. Furthermore, to investigate the potential negative effect of missing labels, we randomly drop a certain proportion of labels in the training set of the Tox21 dataset, and report test results of two representative GNNs, i.e., GCN [11] and GIN [8]. As exemplified in Figure 1(a), the test performance drops sharply when less labels are provided. Existing GNNs handle missing labels by simply ignoring them in the calculation of training loss. As a result, the model performance is impaired for lacking enough supervision information.

There are several challenges to handle missing labels. According to the taxonomy by [12], multi-task learning with missing labels can be regarded as a kind of semi-supervised setting. Although many endeavours have been made for semi-supervised learning, they suffer from two drawbacks in our scenario. At first, they cannot model the co-occurrence relationships between molecules and tasks. As pointed out by [4], the success of multi-task molecular property prediction is that neural networks can "borrow" useful supervision information from **molecules with similar structures** of the other **related tasks**. As a verification, we plot the Pearson correlation coefficients of tasks in the Tox21 dataset in Figure 1(b). The Pearson correlation coefficient is the ratio between the covariance of two variables and the product of their standard deviations. Notably, when we calculate the Pearson correlation coefficients of tasks, we exclude the "None" values. Higher correlation coefficient indicates that the two tasks are more closely related. From the figure, it can be observed that the molecular labels at some tasks are highly correlated. However, it is non-trivial to apply existing semi-supervised learning methods to learn the molecule-molecule similarity, task-task relationships, and molecule-task associations (please refer to Sec. 2.3 for details). Secondly, existing semi-supervised learning methods provide more supervision by annotating unlabeled instances with prediction results, which is prone to inject unreliable information. In multi-task setting, directly using these untrustworthy prediction results is even more risky because an incorrect prediction result for one task may provide misleading supervision information for another related task.

To address the above challenges, we propose to impute missing labels to provide more supervision information. Inspired by [4], we find that the molecule-molecule similarity, task-task relationships, and molecule-task associations can be naturally modeled by a molecule-task bipartite graph, where nodes denote molecules and tasks, and edges denote the labels of molecule-task pairs. In this manner, we cast the goal of imputing missing labels as estimating the missing edges of the bipartite graph, which is illustrated in Figure 2. After that, we devise a graph neural network to learn co-occurrence relationships between molecule-molecule, task-task and molecule-task. The prediction results of this GNN are used for missing label imputation. Considering the imputed pseudo-labels might be untrustworthy, we also propose a certain-first strategy for pseudo-label selection. Our contributions are summarized as follows:

- We highlight the critical importance of considering missing labels when training multi-task GNNs for molecular property prediction.

- We propose a new multi-task GNN framework for molecular property prediction, which imputes missing labels by mining the molecule-task co-occurrence relationships. Besides, we propose a certain-first strategy to select pseudo-labels, which minimize the adverse effect of over-confident labels.
- Extensive results demonstrate the state-of-the-art performance of our proposed method.

2 Related Work

In this section, we review prior work on GNNs for molecular property prediction, multi-task learning, as well as pseudo labeling methods.

2.1 GNNs for Molecular Property Prediction

Recently, GNNs have proved remarkably successful for molecular property prediction. Typically, a molecule could be represented as a graph, where nodes denote atoms, and edges represent chemical bonds. MPNNs [6] with the appropriate message, update, and output functions have a useful inductive bias for predicting molecular properties. MGCN [7] proposes a novel hierarchical GNN which learns the representations of the quantum interactions level by level. GIN [8] theoretically and empirically proves that improving the expressive power of GNNs can boost the molecular property prediction. GraphCL [13] proposes four types of augmentations for general graphs to explor contrastive learning for GNN pre-training. JOAO [14] proposes a unified bi-level optimization framework to dynamically select augmentations in GraphCL. SimGRACE [15] proposes to perturb the encoder in an adversarial way, which introduces less computational overhead while showing better robustness. GraphLoG [16] models the structure of unlabeled graphs at both local- and global-level to assist the molecule representation learning. Apart from the above contrastive learning methods, Meta-MGNN [17] proposes a novel GNN for few shot molecular property prediction by exploring self-supervised learning and meta-learning. Besides, some methods aim to incorporate 3D information for augmenting 2D graph representation learning [18, 19]. Since multi-task molecular property prediction is important in practical applications, most of these above methods have been tested under multi-task setting, which we will elaborate in the next paragraph.

2.2 Multi-task Learning

Multi-task Learning aims to learn multiple tasks simultaneously. It has been proven to improve the performances compared with training under a single task [20], because it is able to utilize the learned knowledge from one task to improve the target task. The simplest approach for multi-task learning is hard parameter sharing [20]. Specifically, the hidden representations are shared across different tasks and only the last prediction layer are distinct for different tasks. To model the relationships of different tasks, some works propose to

use gating mechanism [21, 22]. For example, MMOE [21] explicitly models the task relationships and learns task-specific functionalities by using different gates for different tasks. Existing GNNs [16, 17, 23, 24] usually adopt the hard parameter sharing scheme for the multi-task molecular property prediction. Nevertheless, all these methods ignore the ubiquitous label missing problem, which might hinder the performance. Besides, there are also some multi-task GNNs [25, 26] targeting at node-level tasks, which are outside the scope of our research.

2.3 Semi-supervised Learning for Missing Labels

Multi-task learning with missing labels can be regarded as a kind of semi-supervised setting [12]. Many endeavours have been made for semi-supervised learning, which can be generally grouped into four categories: generative methods, consistency regularization methods, graph-based methods, and pseudo-labeling methods [27]. The success of generative methods and consistency regularization methods depend on the generation of new data [28] and domain-specific augmentations [29], respectively. Nevertheless, it is non-trivial to generate new molecular graphs and conduct molecular augmentations because of the complex topological structure. Besides, graph-based methods [11] usually connect training instances according to their similarities. Then the label information can be propagated from the labeled instances to unlabeled instances. These methods are usually used in a node classification problem and in a single-task setting. Nevertheless, we focus on multi-task graph classification problem. As a result, it is non-trivial to directly utilize graph-based methods in our scenario. Our work is based on pseudo-labeling methods which impute missing labels with pseudo-labels. Meanwhile, it also inherits the advantages of graph-based methods in modeling complex relationships. To our best knowledge, this work is probably the first to develop semi-supervised techniques to boost the performance of GNNs for multi-task molecular property prediction.

3 Preliminary and Related Work

3.1 Problem Description

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X_{\mathcal{V}}, X_{\mathcal{E}})$ represent a molecular graph and let $\mathcal{T}$ represents a task, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, $X_{\mathcal{V}}$ and $X_{\mathcal{E}}$ stand for the matrix for node attributes and edge attributes, respectively. In particular, a node in a molecular graph represents an atom, and an edge represents a chemical bond between two atoms. $D = \{(\mathcal{G}_1, \boldsymbol{Y}_1), \ldots, (\mathcal{G}_{\mathsf{D}}, \boldsymbol{Y}_{\mathsf{D}})\}$ represents the training set, where $\boldsymbol{Y}_i = \left[y_i^1, \ldots, y_i^M\right]$ denotes labels of $\mathcal{G}_i$ over M tasks. Typically, $y_i^j \in \{0, 1\}$ is a binary label, indicating the biological activity (i.e., negative or positive) of $\mathcal{G}_i$ at the j-th task. The multi-task molecular property prediction aims to learn a mapping $f : \mathcal{G} \rightarrow \boldsymbol{Y}$. In this paper, we focus on mitigating the negative effect of missing labels. This is a situation where $\boldsymbol{Y}$

Notation	Description
$\mathbf{h}_v^{(l)}$	embedding of node v resulting from the l-th layer
$\mathbf{z}$	graph embedding
$\hat{y}^j, y^j$	prediction result and the ground truth label for the j-th task
$\mathbf{w}^j$	learnable weights of the prediction layer for the j-th task
$\mathbf{E}_v^{(l)}$	embedding of node v resulting from the l-th layer in the bipartite graph
$\mathbf{W}_{mp}^{(l)}$	weight matrix in the l-th layer of the bipartite graph
$\tilde{y}_i^j$	link prediction result of graph i and task j in the bipartite graph
$\tilde{\boldsymbol{y}}_i^j$	T output values of $\tilde{y}_i^j$
u_i^j	the uncertainty of $\tilde{\boldsymbol{y}}_i^j$
p_i^j	average results of $\tilde{\boldsymbol{y}}_i^j$
τ_p, τ_n	the positive and negative threshold for pseudo-labels
τ	the uncertainty threshold for pseudo-labels

Table 1: Notations used throughout this paper.

contains many "None" values. Notably, missing labels are quite common in existing multi-task molecular benchmarks, such as MoleculeNet [10] and OGB [24].

3.2 Multi-task Learning of GNNs

It is generally known that GNNs are based on the neighborhood aggregation scheme. Specifically, a GNN model iteratively updates the representation of a node by aggregating representations of its adjacent neighboring nodes and edges. By stacking l layers, a node representation $\mathbf{h}_v^{(l)}$ captures the information within its l-hop neighborhoods. Formally, the l-th layer of a GNN generates the representation of a node as follows:

$$\mathbf{h}_v^{(l)} = f_A^{(l)}(\{(\mathbf{h}_v^{(l-1)}, \mathbf{h}_u^{(l-1)}, \mathbf{h}_e^{(l-1)}) \mid e = uv, u \in \mathcal{N}(v)\}), \tag{1}$$

where $\mathbf{h}_v^{(l)}$ denotes the representation of node v at the l-th layer, $h_v^{(0)}$ is initialized by the node attribute X_v, $\mathcal{N}(v)$ is the neighbor set of node v. $\mathbf{h}_e^{(l)} = \mathbf{w}_e^{(l)} X_{uv}$ is the edge embedding resulting from the linear projection of edge attributes. $f_A^{(l)}$ stands for the neighborhood aggregation function at the l-th layer. There have been many architectures for $f_A^{(l)}$, such as GCN [11], GIN [8], and GraphNorm [9]. Without loss of generality, we use GCN and GIN in our model, since they have demonstrated state-of-the-art performance on a variety of tasks. After that, the graph-level embedding is derived from the last layer through an average or a max pooling readout function f_R:

$$\mathbf{z} = f_R(\{h_v \mid v \in \mathcal{V}\}). \tag{2}$$

On top of the graph-level representation $\mathbf{z} \in \mathbb{R}^d$, distinct prediction layers are assigned for each task and generate the prediction results:

$$\hat{y}^j = f_P^j(\mathbf{z}) = \text{sigmoid}(\mathbf{z}^\top \mathbf{w}^j), \tag{3}$$

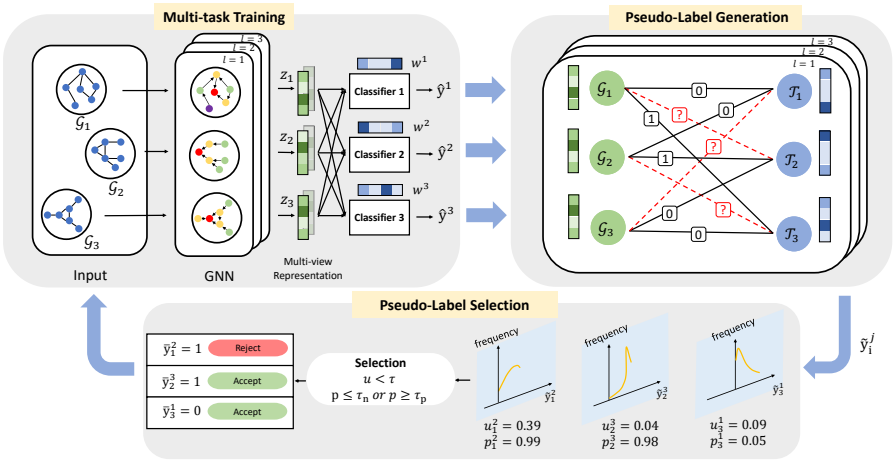


Fig. 2: The workflow of the proposed IM-GNN model, which comprises the following modules: multi-task training, pseudo-label generation and pseudo-label selection. (1) *Multi-task Training*: we conduct supervised-training of GNNs for multi-task molecular property prediction. (2) *Pseudo-label Generation*: a bipartite graph $\mathcal{B}$ is constructed which consists of two types of nodes (i.e., $\mathcal{G}_i$ and $\mathcal{T}_j$), representing molecules and tasks respectively. Node representations are initialized with embeddings from previous module. Black solid lines represent known labels in the training set and red dotted lines denote missing labels. A graph neural network learn the co-occurrence relationships between molecules and tasks, and predict the missing labels. (3) *Pseudo-label Selection*: we estimate the uncertainty of the predictions u_i^j by conducting MC-Dropout. Only the predictions whose uncertainties are lower than τ can be selected as pseudo-labels. τ_p and τ_n are another two thresholds for distinguishing positive and negative pseudo-labels.

where f_P^j denotes the prediction layer, and j is the task index. Usually, the prediction layer is implemented using a linear projection layer followed by a sigmoid function. The vector $\mathbf{w}^j \in \mathbb{R}^d$ denotes the learnable weights of f_P^j. Then the multi-task loss function is defined as the binary cross entropy between the predictions and ground-truth labels:

$$\mathcal{L}_{\text{multi}} = -\frac{1}{\mid D \mid} \sum_{i=1}^{\mid D \mid} \sum_{j=1}^{M} \mathbb{I}\left(y_i^j = y_i^j\right) H\left(y_i^j, \hat{y}_i^j\right), \tag{4}$$

$$H\left(y_i^j, \hat{y}_i^j\right) = [y_i^j log(\hat{y}_i^j) + (1 - y_i^j) log(1 - \hat{y}_i^j)], \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function which excludes the calculation of "None" values (i.e., missing labels), and $\mid D \mid$ is the number of graphs in the training set.

4 Proposed Method

The key idea behind our method is that we conduct missing label IMputation for multi-task GNNs by mining the complex molecule-task co-occurrence relationships. Therefore, our model is called IM-GNN for brevity. The workflow of IM-GNN is illustrated in Figure 2. Below we first present the multi-task training procedure of GNNs for molecular property prediction, followed by the graph-based pseudo-label generation process. Finally, we elaborate on the details of pseudo-label selection.

4.1 Multi-view Representation for Multi-task Training

Inspired by [30], we assume that a single learning task is sensitive to multiple characteristics of the graph and that only one shared graph representation may limit the performance at some tasks. As a result, instead of adopting the widely used average pooling or max pooling operation (formulated in Eq. (2)), we propose to extract multi-view graph representations for different tasks. This can be implemented through multi-head attention [31]:

$$\mathbf{z} = \text{concatenation}(\text{head}_1, \ldots, \text{head}_K), \tag{6}$$

where,

$$\text{head}_k = \sum_{v=1}^{N} \text{softmax}(\mathbf{w}_{att}^k \mathbf{h}_v^k). \tag{7}$$

Here, K is the number of heads. $\mathbf{w}_{att}^k$ and $\mathbf{h}_v^k$ denote the attention weights and node representation of the k-th view, respectively. Then, the concatenated multi-view graph representation is fed into the task classifier (formulated in Eq. (3)) to obtain the prediction result.

4.2 Graph-based Pseudo-Label Generation

4.2.1 Graph construction

We conduct missing label imputation by mining the co-occurrence relationships between molecular graphs and tasks. Specifically, we first construct a molecule-task bipartite graph $\mathcal{B}$. As illustrated in top right part of Figure 2, each molecule is represented as a green node and each task is represented as a blue node. Therefore, there are $(\mathsf{D}+M)$ nodes in $\mathcal{B}$ in total. Each edge $(\mathcal{G}_i, \mathcal{T}_j)$ (illustrated as a solid line) means that the graph instance $\mathcal{G}_i$ has a label y_i^j at task $\mathcal{T}_j$. The red dotted lines denote the missing labels in the training set. In this manner, the imputation of missing labels can be cast into edge prediction in $\mathcal{B}$.

If node representations in $\mathcal{B}$ are randomly initialized, the structure and attribute information of molecules $\{\mathcal{G}_1, \ldots, \mathcal{G}_{\mathsf{D}}\}$ cannot be utilized, which might probably degrade the performance. To this end, we use the learnt graph representation $\mathbf{z}$ in Eq. (6) to initialize the corresponding molecule node in $\mathcal{B}$. To initialize the task node, we adopt the learnt weights w^j of f_P^j, since the

prediction layer generates predictions by calculating the inner product of $\mathbf{z}$ and w^j, which is formulated in Eq. (3). In other words, w^j is task-specific and therefore it can be used to represent the j-th task.

4.2.2 Message propagation

Since the bipartite graph $\mathcal{B}$ serves as a bridge connecting the molecular graphs and tasks, it is necessary to make the information flow to learn these molecule-task co-occurrence relationships. In view of this, we propose to use graph neural networks, which can facilitate the message propagation process. Specifically, one node in $\mathcal{B}$ can aggregate information from its 1-hop neighborhood, which is formulated as:

$$\mathbf{E}_v^{(l)} = \sigma\Big(\sum_{u\in\mathcal{N}(v)} \tilde{\mathbf{A}}_{uv}\mathbf{W}_{mp}^{(l)}\mathbf{E}_u^{(l-1)}\Big), \tag{8}$$

where $\mathbf{E}_v^{(l)}$ is the embedding of node v resulting from the l-th layer, $\mathcal{N}(v)$ is a set containing the neighbors of node v and itself, $\tilde{\mathbf{A}}$ is the normalized adjacency matrix of $\mathcal{B}$, $\mathbf{W}_{mp}^{(l)}$ is a trainable weight matrix, and $\sigma(x) = \max(0, x)$ is the ReLU [32] activation function.

4.2.3 Model optimization and pseudo-label generation

After propagating several layers, the long-range co-occurrence relationships can be captured. Particularly, the 1-hop connectivity encodes the associations between molecules and tasks; the second-hop connectivity encodes the molecule-molecule similarity and the task-task correlation. Besides, the high-order connectivity can propagate the information from multi-hop neighbors. Based on the node embedding resulting from the last layer, we conduct the inner product to estimate the edge labels of $\mathcal{B}$:

$$\tilde{y}_i^j = \text{sigmoid}(\mathbf{E}_{\mathcal{G}_i}^\top \mathbf{E}_{\mathcal{T}_j}). \tag{9}$$

Finally, we adopt the binary cross entropy loss to optimize the model parameters:

$$\mathcal{L}_{\text{IM-GNN}} = -\frac{1}{\mathsf{D}}\sum_{i=1}^{\mathsf{D}}\sum_{j=1}^{M}\mathbb{I}\left(y_i^j = y_i^j\right) H\left(y_i^j, \tilde{y}_i^j\right). \tag{10}$$

Once the above network is optimized, we are able to predict the labels of missing edges based on Eq. (9). The predicted results can be used for missing label imputation.

4.3 Uncertainty-Aware Pseudo-Label Selection

It is crucial to select pseudo-labels as accurately as possible since wrong pseudo-labels severely degrade the model performance. In the multi-task setting, reliable pseudo-labels are even more important to avoid unexpected

Algorithm 1 IM-GNN Training Algorithm

for stage $\leftarrow 1, 2, \cdots$ **do**
  % Step 1: Multi-task Training
  Random initialize (or using the pre-trained) multi-task GNN and train it for a fixed number of epochs with Eq. (4).
  Obtain embeddings of molecules and tasks.
  % Step 2: Pseudo-label generation
  Initialize the node representations of $\mathcal{B}$ with the embeddings in step 4.
  Train the GNN on the bipartite graph $\mathcal{B}$ for a fixed number of epochs with Eq. (9).
  % Step 3: Pseudo-label selection
  Conduct MC-dropout T times using Eq. (8) and obtain $\tilde{\boldsymbol{y}}_i^j$.
  Obtain the uncertainty u_i^j and the final prediction result p_i^j with Eq. (10) and Eq. (11).
  Select pseudo-labels with Eq. (12).
  Impute missing labels $\boldsymbol{Y}$ with the selected pseudo-labels.
end for
return the desired multi-task mapping $f : \mathcal{G} \rightarrow \boldsymbol{Y}$.

propagation of misleading supervision information between tasks. Nevertheless, prior studies [33, 34] show that deep learning models suffer from over-confident predictions. As a result, it is inappropriate to directly select pseudo-labels according to prediction results. To pick up accurate pseudo-labels and prevent error from being propagated, we propose a certain-first strategy for pseudo-label selection, which is illustrated in the bottom part of Figure 2.

4.3.1 Uncertainty estimation

We estimate the uncertainty of pseudo-label $\tilde{y}_i^j$ via MC-dropout [33]. In particular, we conduct T forward passes with dropout layers activated at test time. Then we obtain T output values $\tilde{\boldsymbol{y}}_i^j = [\tilde{y}_i^{j(1)}, \cdots, \tilde{y}_i^{j(T)}]$. According to [33], the uncertainty of $\tilde{y}_i^j$ can be measured by the variance of $\tilde{\boldsymbol{y}}_i^j$. Since $\tilde{y}_i^j$ is a binary categorical variable, we adopt Shannon's entropy to model the variance of $\tilde{y}_i^j$ as:

$$u_i^j = -\frac{1}{T}\sum_{t=1}^{T} \tilde{y}_i^{j(t)} \log(\tilde{y}_i^{j(t)}) + \left(1 - \tilde{y}_i^{j(t)}\right) \log\left(1 - \tilde{y}_i^{j(t)}\right), \tag{11}$$

where u_i^j denotes the uncertainty. Besides, we use p_i^j to represent the average results of $\tilde{\boldsymbol{y}}_i^j$:

$$p_i^j = \frac{1}{T}\sum_{t=1}^{T} \tilde{y}_i^{j(t)}. \tag{12}$$

4.3.2 Certain-first selection

The uncertainty indicates how confident a model is with respect to the prediction results. The smaller the uncertainty, the more confident the prediction. So we select the predictions of the missing edges according to their uncertainty values. Specifically, we set a threshold τ, and only the predictions whose uncertainties are lower than τ can be selected as pseudo-labels. For example, as illustrated in Figure 2, we plot the distributions of $\tilde{y}_i^j$, where "fat" and "thin" distributions represent high and low uncertainties, respectively. p_1^2 is rejected because its uncertainty is high. This showcases the ability of IM-GNN to exclude over-confident pseudo-labels. Besides, we assign another two thresholds, i.e., τ_p and τ_n, to distinguish positive and negative pseudo-labels. In summary, the pseudo-labels are selected as follows:

$$\bar{y}_i^j = \begin{cases} 1 & \text{if } u_i^j < \tau \text{ and } p_i^j \geq \tau_p \\ 0 & \text{if } u_i^j < \tau \text{ and } p_i^j \leq \tau_n \\ \text{reject} & \text{otherwise} \end{cases}. \tag{13}$$

Finally, the selected pseudo-labels are used to impute missing labels in $\boldsymbol{Y}$, which provide more reliable supervision information for the multi-task training of GNNs. As a result, the proposed IM-GNN can be regarded as a semi-supervised learning method. The overall training algorithm is summarized in Algorithm 1.

4.4 Complexity Analysis

Compared to vanilla multi-task GNNs, the additional computation complexity of IM-GNN comes from the pseudo-label generation process. Suppose there are m labels in the training set, then the bipartite graph contains $(\mathsf{D}+M)$ nodes and m edges. Correspondingly, the complexity of the message propagation is $\mathcal{O}((\mathsf{D}+M)md)$. In other words, the complexity is proportional to the size of training instances and training labels.

5 Experiments

In this section, we conduct experiments to evaluate our model through answering the following research questions.

- **RQ1**. How does the proposed IM-GNN perform compared with existing state-of-the-art GNNs on multi-task molecular property prediction?
- **RQ2**. How does IM-GNN perform in imputing missing labels?
- **RQ3**. How do different components affect the performance?
- **RQ4**. How do key hyper-parameters impact the model performance?

5.1 Experimental Setup

5.1.1 Datasets and learning protocol

We adopt five widely used multi-task datasets in MoleculeNet [10] for molecular property prediction, which covers a wide range of molecular tasks such as response in bioassays, toxicity and adverse reactions:

- **Muv [35]:**. A dataset specifically designed for validation of virtual screening techniques.
- **Toxcast [36] & Txo21** [1]**:**. Two datasets containing qualitative toxicity measurements on different targets, such as nuclear receptors and stress response pathways.
- **Clintox [37]:**. A dataset compares drugs approved by the FDA (i.e., Food and Drug Administration) and drugs that have failed clinical trials for toxicity reasons.
- **Sider [38]:**. A dataset contains the adverse drug reactions of FDA approved drugs.

The raw data of molecules are given as *SMILES* strings, so we adopt the graph features processed by [24] and [23]. Specifically, we conduct experiments in the settings of supervised learning [24] and transfer learning [23]. (1) In supervised learning, we follow the protocol in [24], where we train the model from scratch. (2) In transfer learning, we closely follow the protocol in [23], where we pre-train on a larger dataset then finetune and evaluate on the above datasets using the given training/validation/test split. We use the scaffold split with the ratio of 80%/10%/10% for training, validation, and test set, respectively.

Although the above datasets in [23] and [24] share the same name, they are different in node features. As shown in [24], the authors use additional atom features such as formal charge and whether the atom is in the ring, which are not adopted in the previous work [23]. The dataset statistics are summarized in Table 2. It is obvious that missing labels are quite common in these multi-task datasets, especially for large-scale datasets. The reason is that collecting complete labels for large-scale datasets requires more time and cost. Notably, the original version of the Clintox and Sider dataset, which are smallest in Table 2, contain **complete** training labels. In this paper, we randomly drops 80% training labels (**marked by** $*$) to simulate the label-missing scenario in real-life applications. Since multi-task learning with missing labels ca be regarded as a kind of semi-supervised setting [12], the two learning protocols can be used to demonstrate the effectiveness of the proposed method.

We run each model ten times with random seed ranging from 0 to 9. And we report the mean and standard deviation of test ROC-AUC across all tasks. In particular, as stated in [24], Average Precision (AP) is a more appropriate metric for heavily-imbalanced data, we report the AP score on Muv dataset in supervised learning protocol for more intuitive comparison.

[1] https://tripod.nih.gov/tox21/challenge/

5.1.2 Compared Algorithms

Training from scratch. The naive baseline training methods from random initialization are compared. Here, we select two representative GNN models, GCN [11] & GIN [8] as backbones, which are widely used for molecular property prediction [24]. They can effectively capturing the structure and attribute information of graphs. Besides, although GraphNorm [9] is designed for accelerating the training process of GNNs, it shows better generalization performance for molecular property prediction. We also regard GraphNorm as a strong baseline.

Self-supervised learning methods for transfer learning. We compare IM-GNN with state-of-the-art graph self-supervised learning methods: EdgePred [23], AttMasking [23], ContextPred [23], InfoGraph [39], GraphCL [23] and GraphLoG [16]. For the implementation of IM-GNN, we apply GraphLoG as the base model.

Other state-of-the-art baselines. Since our work is related to multi-task learning (MTL) and semi-supervised learning, we also compare strong MTL and semi-supervised learning methods. For MTL, we select MMOE [21] and GradNorm [40] as baselines. MMOE adopts multiple classifiers and learns task-specific functionalities by using different gates for different tasks. GradNorm tunes loss weights in a multi-task learning setting based on balancing the training rates of different tasks. For semi-supervised learning, the VAnilla Self-Training [41] (VA-ST) is compared. VA-ST selects the unlabeled instances with high confidence as training targets, which provide more supervision information for the model. Specifically, it can be regarded as the model variant of IM-GNN without the bipartite graph part and the uncertainty selection part. It directly selects pseudo-labels from the prediction of multi-task GNNs.

5.1.3 Implementation details

For transfer learning, we closly follow the setting in [16]. For fair comparison, we use the data sets as in [23]. Specificically, we apply a subset of ZINC15 database[42], which contains 2 million unlabeled molecules. For the pre-training strategy, we also use GraphLog[16] as the base model. We adopt a five-layer GIN with 300-dimensional hidden units for all compared methods, including MMOE, GradNorm, VA-ST and IM-GNN. We use a linear classifier for fine-tuning and adopt an Adam optimizer(learning rate: 0.001). Unless otherwise specified, the batch size N is set as 512, and the hierarchical prototypes' depth L_p is set as 3.

For supervised training from scratch, We use a five-layer architecture for GCN and GIN. Specifically, the number of sub-layers in MLP is set to 2 for GIN. For GraphNorm, we adopt normalization strategy on each layer. For MMOE, we use 5 experts/classifiers for classification. For the proposed IM-GNN, we use GCN and GIN as base GNN methods for multi-task learning.

We use a two-layer GCN for edge prediction in the bipartite graph. For the setting of other hyper-parameters, we train all multi-task GNNs 100 epochs

Dataset	#Tasks	#Graphs	Avg. #Nodes	Missing Labels
Muv	17	93,087	24.2	84%
Toxcast	617	8,576	18.8	71%
Tox21	12	7,831	18.6	17%
Clintox	2	1477	26.2	80%
Sider	27	1427	33.6	80%

Table 2: Statistics of the molecular graph datasets.

Method	Muv	Toxcast	Tox21	Clintox	Sider	Avg
No Pre-train	71.8 ± 2.5	63.4 ± 0.6	74.0 ± 0.8	54.5 ± 4.2	52.4 ± 1.6	63.2
EdgePred	74.1 ± 2.1	64.1 ± 0.6	76.0 ± 0.6	59.7 ± 3.8	55.6 ± 0.7	65.9
InfoGraph	75.6 ± 1.2	63.1 ± 0.3	75.5 ± 0.6	67.8 ± 1.9	54.1 ± 1.1	67.2
AttrMasking	74.7 ± 1.4	64.2 ± 0.5	76.7 ± 0.4	69.4 ± 4.1	56.6 ± 0.8	68.3
ContextPred	75.8 ± 1.7	63.9 ± 0.6	75.7 ± 0.7	63.5 ± 3.6	56.3 ± 0.6	67.0
GraphCL	74.5 ± 1.3	63.8 ± 0.4	75.4 ± 0.9	67.2 ± 1.8	56.2 ± 0.6	67.4
GraphLoG	76.0 ± 1.1	63.5 ± 0.7	75.7 ± 0.5	72.8 ± 3.1	56.8 ± 1.2	69.0
MMOE	73.5 ± 1.1	63.8 ± 0.6	75.9 ± 0.6	73.5 ± 2.8	57.6 ± 0.9	68.9
GradNorm	74.1 ± 1.2	63.9 ± 0.7	75.8 ± 0.8	74.2 ± 2.7	58.2 ± 0.8	69.2
VA-ST	76.8 ± 1.2	64.0 ± 0.7	76.0 ± 0.7	74.1 ± 3.2	58.5 ± 1.2	69.9
IM-GNN	**77.2 ± 1.1**	**64.8 ± 0.6**	**76.8 ± 0.6**	**75.4 ± 2.7**	**60.0 ± 1.1**	**70.8**

Table 3: Summary of performance (%) on molecular property prediction benchmarks under transfer learning. The rightmost column reports the average performance of each method on different datasets. The highest performance is highlighted in bold.

with an Adam SGD optimizer [43]. The learning rate is set to 0.001 and the dropout [44] rate is set to 0.5. For the training of GCN in the bipartite graph, we run 400 epochs with the learning rate 0.005. For MC-droput, we set the dropout rate to 0.5. We select 20% pseudo-labels that have the lowest uncertainties in each stage. In other words, the threshold τ is set to 20 percentile of the uncertainties. Besides, the threshold for τ_p and τ_n are selected by grid search in $\{0.85, 0.86, \ldots, 0.99\}$ and $\{0.05, 0.1, \ldots, 0.3\}$ respectively. All models are implemented using Pytorch on a computer server with four NVIDIA Tesla V100S GPUs (32GB memory each).

5.2 Overall Performance (RQ1)

Transfer learning results. In Table 3, we report the transfer learning results of different GNN methods. They are initialized by pre-trained weights, and then fine-tuned on each specific dataset. It can be observed that GIN without pre-training performs bad on all datasets. And GraphLoG achieves satisfactory results among self-supervised methods. Using the self-supervised learning

Method	Muv	Toxcast	Tox21	Clintox	Sider	Avg
GCN	11.4 ± 2.9	63.5 ± 0.4	75.3 ± 0.7	87.6 ± 1.7	57.1 ± 1.5	59.0
GCN + GraphNorm	6.5 ± 2.8	63.7 ± 0.5	74.2 ± 0.6	81.5 ± 2.8	58.3 ± 1.6	56.9
GCN + MMOE	11.4 ± 2.6	63.9 ± 0.8	75.5 ± 0.7	86.6 ± 2.0	58.6 ± 1.3	59.2
GCN + GradNorm	11.6 ± 2.7	64.0 ± 0.6	75.8 ± 0.6	87.2 ± 2.2	59.2 ± 1.4	59.6
GCN + VA-ST	12.4 ± 2.6	64.1 ± 0.5	75.8 ± 0.4	89.0 ± 1.5	57.9 ± 0.8	59.8
IM-GCN	**13.8 ± 2.0**	**64.8 ± 0.5**	**76.3 ± 0.4**	**90.7 ± 0.9**	**60.4 ± 0.8**	**61.2**
GIN	8.8 ± 2.1	63.4 ± 0.7	74.9 ± 0.5	83.3 ± 3.1	57.9 ± 1.6	57.7
GIN + GraphNorm	5.2 ± 2.26	64.5 ± 0.7	74.0 ± 0.4	81.1 ± 3.6	58.1 ± 1.8	56.6
GIN + MMOE	9.2 ± 2.3	62.7 ± 0.7	75.0 ± 0.6	78.8 ± 3.9	58.4 ± 1.0	56.8
GIN + GradNorm	9.4± 2.1	63.2 ± 0.6	75.1 ± 0.4	80.2 ± 3.5	58.6 ± 1.9	57.3
GIN + VA-ST	9.3 ± 2.4	63.9 ± 0.6	75.2 ± 0.5	82.3 ± 2.8	59.0 ± 0.8	57.9
IM-GIN	**10.2 ± 2.5**	**64.6 ± 0.5**	**75.7 ± 0.5**	**85.3 ± 2.8**	**60.1 ± 0.6**	**59.2**

Table 4: Summary of performance (%) on molecular property prediction benchmarks under training from scratch setting. The compared numbers of GCN and GIN are from [24]. The rightmost column reports the average performance of each method on different datasets. The highest performance is highlighted in bold.

strategies of GraphLoG, we notice that GradNorm usually performs better than MMOE for multi-task molecular property prediction. This can be attributed to the ability of GradNorm in balancing different tasks. Besides, we observe that VA-ST outperforms GradNorm and MMOE in most cases. We assume that the imputation of missing labels provide more supervision information and can boost the performance. Last but not least, the proposed IM-GNN outperforms other methods on all datasets. This verifies the effectiveness of the proposed method. Specifically, even compared with VA-ST, IM-GNN shows satisfactory improvement. We deem this improvement is mainly from the co-occurrence relationship modeling and uncertainty-aware pseudo-label selection mechanism.

Supervised learning results. Under the setting of training from scratch, the overall performances of all compared models are reported in Table 4. In all, it is apparent that IM-GNN outperforms all the other methods by achieving the best average performance with 61.2% for GCN-based models and 59.2% for GIN-based models. Moreover, we have the following observations: (1) GraphNorm shows unstable performance on different datasets. It achieves improvement on Toxcast and Sider datasets while the results on the other three datasets are unsatisfactory. It may be because the normalization operation discards useful information for some datasets. (2) Compared with the vanilla GCN, the two MTL methods, i.e., MMOE and GradNorm, show favorable performances on most of the datasets (except for Clintox). Nevertheless, it cannot solve the label missing issue, restricting the performance. (3) VA-ST obtains stable improvements on all datasets, indicating that the imputation of missing labels can provide more useful supervision information. (4) Compared to other approaches, IM-GNN achieves consistent improvements on all datasets in terms of GCN and GIN. Specifically, it outperforms the base model GCN

by margins of 2.4%, 1.3%, 1.03%, 3.1%, and 3.3% on different datasets. The same trend holds for IM-GIN with its base model GIN as well, which demonstrates the effectiveness and compatibility of our proposed approach. Even compared with the strongest baseline self-training, IM-GNN achieves about 1% improvement on average. Notably, the improvement on Tox21 is relatively lower than those on other datasets. This may mainly because there are only 17% missing labels on Tox21, and the imputation cannot provide too much useful supervision information.

Please kindly note that the numbers in these two tables cannot be directly compared, because the node features are different under these two settings. Besides, for the ease of comparison with previous work [16, 24], we report the ROC-AUC result for MUV in Table 3 while showing the Average Precision result in Table 4. Please refer to Section 5.1.1 for more details. In all, the results in above two tables show that IM-GNN achieves stable improvements on all datasets, verifying the effectiveness of the proposed method.

5.3 Imputation Accuracy (RQ2)

The key idea of IM-GNN is to provide more supervision information via missing label imputation. So we also evaluate how accurate are the imputed pseudo-labels. Since there is no ground-truth for missing labels, we specifically mask 50% training labels of Tox21 and Sider by random. Then, we train IM-GNN (without masked labels) to predict these masked labels. The results are presented in Figure 3 and Figure 4. In particular, the horizontal axis refers to how many masked labels are evaluated. We have the following two observations.

Firstly, the imputation results of each model show a clear decline trend. This is related to the strategy used for pseudo-label selection. Specifically, we give priority to evaluate the pseudo-labels with low uncertainties and high confidences. As a result, when more pseudo-labels (with relatively high uncertainties) are evaluated, more false predictions are likely to be contained, degrading the performance. Secondly, when the pseudo-labeling rate equals to 0.4, the imputation ROC-AUC are almost 90%. This result also proves that IM-GNN can provide enough and reliable supervision information.

5.4 Ablation Studies (RQ3)

To demonstrate the effectiveness of the proposed multi-view graph representation, molecule-task-based GNN, and the certain-first strategy, we conduct ablation studies by removing these three modules respectively. In the last subsection, we have observed similar performance of IM-GNN under two learning protocols (i.e., transfer learning and supervised learning) and using different network backbones (i.e., GCN and GIN). Without loss of generality, here we use GCN as the network backbone and report the results under supervised learning protocol. The results are shown in Table 5. We compare the following four model variants.

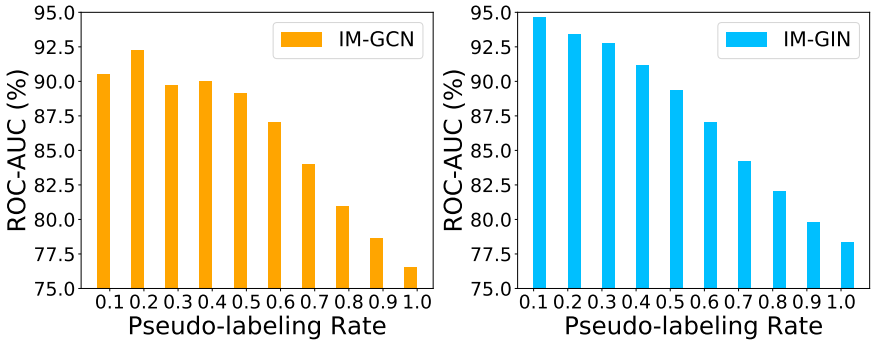


Fig. 3: Imputation performance on Tox21.

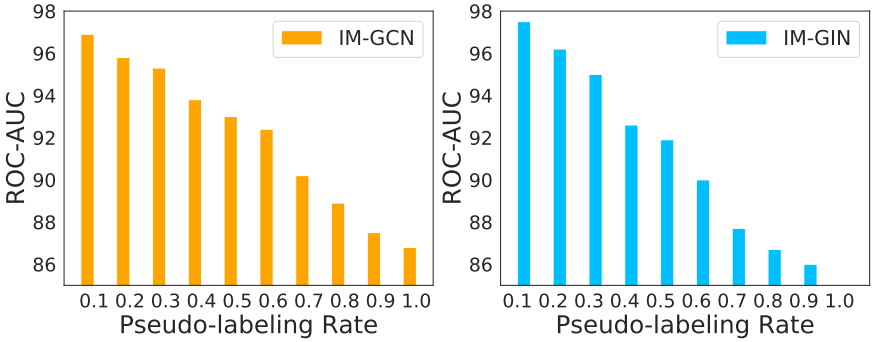


Fig. 4: Imputation performance on Sider.

IM-GNN w/o multi-view. Compared with IM-GNN, this model uses only one view representation. In other words, it is the degeneration variant of IM-GNN where the head number equals one in Eq. (6).

GNN+ZeroImp. For this model, we subtitute the graph-based pseudo-label generation part and the uncertainty-aware pseudo-label selection scheme with zero imputation. Specifically, since most of the labels of the training set are negative, we simply use 0 for imputation and explore the MTL results in this case.

GNN+VA-ST. For this model, pseudo-labels are directly taken from the multi-task GNNs, which is formulated in Eq. (3). In other words, the Graph-based pseudo-label generation part (which is depicted in Sec. (4.2)) of IM-GNN is removed.

IM-GNN w/o uncertainty. Compared with our proposed method, this model uses only τ_p and τ_n to distinguish positive and negative pseudo-labels. It does not consider the uncertainty of pseudo-labels.

IM-GNN. It denotes our proposed model.

Method	Muv	Toxcast	Tox21	Clintox	Sider	Avg
GNN	11.4 ± 2.9	63.5 ± 0.4	75.3 ± 0.7	87.6 ± 1.7	57.1 ± 1.5	59.0
w/o multi-view	13.4 ± 2.5	64.2 ± 0.4	75.9 ± 0.4	88.7 ± 2.1	59.5 ± 0.8	60.3
GNN+ZeroImp	11.2 ± 2.6	63.8 ± 0.4	74.6 ± 0.3	86.5 ± 1.9	56.5 ± 0.9	58.5
GNN+VA-ST	12.4 ± 2.3	64.1 ± 0.4	75.8 ± 0.6	88.1 ± 2.1	57.9 ± 1.3	59.7
w/o uncertainty	13.3 ± 2.7	64.2 ± 0.5	76.0 ± 0.4	89.6 ± 1.8	59.6 ± 1.4	60.5
IM-GNN	**13.8 ± 2.0**	**64.8 ± 0.5**	**76.3 ± 0.4**	**90.7 ± 0.9**	**60.4 ± 0.8**	**61.2**

Table 5: Performance (%) of model variants on molecular property prediction. The bold number denotes the best performance.

Compared with IM-GNN, it can be observed that removing any module results in performance degradation, which verifies the effectiveness of each module. (1) We can see that removing multi-view graph representation results in 0.9% performance degradation. This suggests that the proposed multi-head attention mechanism is helpful for multi-task molecular property prediction. (2) The simple imputation scheme, zero imputation, hurts the performance of vanilla GNN. We assume that this kind of imputation introduces much label noise to the model. (3) Compared with *GNN+ZeroImp*, *GNN+VA-ST* achieves superior performance because of its pseudo-labels are more reliable. However, apart from *GNN+ZeroImp*, *GNN+VA-ST* shows the worst performance among these model variants. We can see that removing the bipartite graph brings 1.5% performance degradation. This demonstrates that importance of modeling the co-occurrence relationships between molecules and tasks. (4) Besides, comparing IM-GNN and *IM-GNN w/o uncertainty*, we observe that the uncertainty selection mechanism brings further improvement.

5.5 Sensitivity Analysis

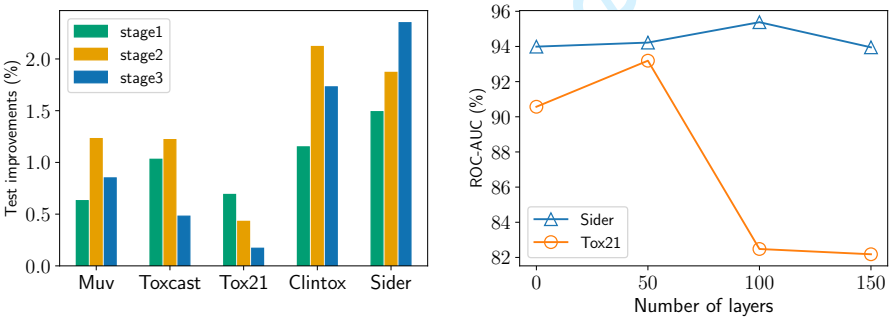


Fig. 5: Sensitivity studies of (a) the number of self-training stages. (b) the number of GNN layers K for the molecule-task graph.

5.5.1 Influence of stage number

In this subsection, we perform sensitivity analysis on critical hyper-parameters of IM-GNN, namely the stage number and layer number of the bipartite graph.

The proposed IM-GNN works in a multi-stage self-training framework, where a stage refer to a for-loop in Algorithm 1. So we investigate how the stage number influence the multi-task learning performance. Specifically, we use GCN as base GNN and plot the improvement of IM-GCN over GCN in Figure 5(a). It can be observed that IM-GCN obtains the best result at the second stage for most datasets, such as Muv, Toxcast, and Clintox. This fact demonstrates the importance of multi-stage training. Since more supervision information is provided in the next-stage, the multi-task GNNs produce better embeddings for molecules and tasks, which help to generate more useful pseudo-labels. Meanwhile, more stages will inevitably introduce more noisy labels, degrading the overall performance. Besides, we observe IM-GCN obtains the best performance at the first and third stage on Tox21 and Sider, respectively. We suppose this result is mainly related to the ratio of missing labels. There are only 17% missing labels in Tox21 and 80% missing labels in Sider. For the dataset with more missing labels, IM-GCN can provide more useful supervision information in the last few stages.

5.5.2 Influence of layer number

The number of layers of the GNN for bipartite graph $\mathcal{B}$ is also a critical hyper-parameter, which decides how many hops of neighbor information are propagated. In this part, we study the imputation performance of IM-GCN by varying the GNN layers on $\mathcal{B}$ from one to four. Specifically, we randomly mask 50% labels of Tox21 and 80% labels of Sider. We set τ to 20 percentile of the uncertainties and set . The results are shown in the left part of Figure 5(b). It can be observed that the performances on Tox21 and Sider share the same trend. To be more specific, the performance improves with the number of layers at first, which demonstrates modeling the 2-hop neighborhood information (i.e., molecule-molecule similarity and task-task correlation) is beneficial for the imputation. Nevertheless, too many layers will inevitably introduce redundant parameters and over-smoothing [45] to the model, leading to degraded performances as well.

6 Conclusion

In this paper, we have investigated the adverse effect of missing labels for multi-task molecular property prediction. We regard this scenario as a semi-supervised learning problem and develop semi-supervised techniques to boost performance. We first propose multi-view graph representation for multi-task training. Besides, to provide more useful supervision information, we propose a new framework IM-GNN, which imputes missing labels by mining the collaborative molecule-task relationships. The key of IM-GNN is the proposed GNN network for the molecule-task bipartite graph. By propagating information in

this bipartite graph, the molecule-molecule similarity, task-task relationships, and molecule-task associations can be learned. As a result, the missing label of a molecule-task pair can be imputed based on the labels of its similar molecule-task pairs. After that, we select pseudo-labels according to the uncertainty of the prediction to alleviate the negative effect of noisy labels. At last, the selected pseudo-labels are leveraged to refine the multi-task training process of GNNs. Experimental results validate the effectiveness of the proposed method.

Meanwhile, we have noticed that there are other kind of semi-supervised methods showing promising results, such as regularization methods [46]. However, domain-specific augmentations are usually needed and appropriate molecular augmentations remains to be explored. As a result, other semi-supervised learning methods are worth exploring in the future work. Moreover, since the proposed IM-GNN is model-agnostic, it is prospective to be combined with other semi-supervised learning methods for this scenario.

References

[1] Stephenson, N., Shane, E., Chase, J., Rowland, J., Ries, D., Justice, N., Zhang, J., Chan, L., Cao, R.: Survey of machine learning techniques in drug discovery. Current drug metabolism (2019)

[2] Shen, J., Nicolaou, C.A.: Molecular property prediction: recent trends in the era of artificial intelligence. Drug Discovery Today: Technologies (2019)

[3] Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V.: Massively multitask networks for drug discovery. arXiv preprint arXiv:1502.02072 (2015)

[4] Xu, Y., Ma, J., Liaw, A., Sheridan, R.P., Svetnik, V.: Demystifying multitask deep neural networks for quantitative structure–activity relationships. Journal of chemical information and modeling (2017)

[5] Sadawi, N., Olier, I., Vanschoren, J., Van Rijn, J.N., Besnard, J., Bickerton, R., Grosan, C., Soldatova, L., King, R.D.: Multi-task learning with a natural metric for quantitative structure activity relationship learning. Journal of Cheminformatics (2019)

[6] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning (2017)

[7] Lu, C., Liu, Q., Wang, C., Huang, Z., Lin, P., He, L.: Molecular property prediction: A multilevel quantum interactions modeling perspective. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)

[8] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019)

[9] Cai, T., Luo, S., Xu, K., He, D., Liu, T.-y., Wang, L.: Graphnorm: A principled approach to accelerating graph neural network training. In: International Conference on Machine Learning (2021)

[10] Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: Moleculenet: a benchmark for molecular machine learning. Chemical science (2018)

[11] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)

[12] Zhou, Z.-H.: A brief introduction to weakly supervised learning. National science review (2018)

[13] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems (2020)

[14] You, Y., Chen, T., Shen, Y., Wang, Z.: Graph contrastive learning automated. In: International Conference on Machine Learning (2021)

[15] Xia, J., Wu, L., Chen, J., Hu, B., Li, S.Z.: Simgrace: A simple framework for graph contrastive learning without data augmentation. In: Proceedings of the ACM Web Conference 2022 (2022)

[16] Xu, M., Wang, H., Ni, B., Guo, H., Tang, J.: Self-supervised graph-level representation learning with local and global structure. In: International Conference on Machine Learning (2021)

[17] Guo, Z., Zhang, C., Yu, W., Herr, J., Wiest, O., Jiang, M., Chawla, N.V.: Few-shot graph learning for molecular property prediction. In: Proceedings of the Web Conference 2021 (2021)

[18] Liu, S., Wang, H., Liu, W., Lasenby, J., Guo, H., Tang, J.: Pre-training molecular graph representation with 3d geometry. In: International Conference on Learning Representations (2021)

[19] Stärk, H., Beaini, D., Corso, G., Tossou, P., Dallago, C., Günnemann, S., Liò, P.: 3d infomax improves gnns for molecular property prediction. In: International Conference on Machine Learning (2022)

[20] Caruana, R.: Multitask learning. Machine learning (1997)

[21] Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018)

[22] Tang, H., Liu, J., Zhao, M., Gong, X.: Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In: Fourteenth ACM Conference on Recommender Systems (2020)

[23] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J.: Strategies for pre-training graph neural networks. In: International Conference on Learning Representations (2020)

[24] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. In: Advances in Neural Information Processing Systems (2020)

[25] Avelar, P., Lemos, H., Prates, M., Lamb, L.: Multitask learning on graph neural networks: Learning multiple graph centrality measures with a unified network. In: International Conference on Artificial Neural Networks (2019)

[26] Guo, P., Deng, C., Xu, L., Huang, X., Zhang, Y.: Deep multi-task augmented feature learning via hierarchical graph neural network. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases (2021)

[27] Yang, X., Song, Z., King, I., Xu, Z.: A survey on deep semi-supervised learning. arXiv preprint arXiv:2103.00550 (2021)

[28] Odena, A.: Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583 (2016)

[29] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in Neural Information Processing Systems (2019)

[30] He, J., Lawrence, R.: A graphbased framework for multi-task multi-view learning. In: International Conference on Machine Learning (2011)

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)

[32] Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning (2010)

[33] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning (2016)

[34] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in Neural Information Processing Systems (2017)

[35] Rohrer, S.G., Baumann, K.: Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. Journal of chemical information and modeling (2009)

[36] Richard, A.M., Judson, R.S., Houck, K.A., Grulke, C.M., Volarath, P., Thillainadarajah, I., Yang, C., Rathman, J., Martin, M.T., Wambaugh, J.F., et al.: Toxcast chemical landscape: paving the road to 21st century toxicology. Chemical research in toxicology (2016)

[37] Gayvert, K.M., Madhukar, N.S., Elemento, O.: A data-driven approach to predicting successes and failures of clinical trials. Cell chemical biology (2016)

[38] Kuhn, M., Letunic, I., Jensen, L.J., Bork, P.: The sider database of drugs and side effects. Nucleic acids research (2016)

[39] Sun, F.-Y., Hoffman, J., Verma, V., Tang, J.: Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In: International Conference on Learning Representations (2020)

[40] Chen, Z., Badrinarayanan, V., Lee, C.-Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: International Conference on Machine Learning (2018)

[41] Lee, D.-H., *et al.*: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML (2013)

[42] Sterling, T., Irwin, J.J.: Zinc 15–ligand discovery for everyone. Journal of chemical information and modeling **55**(11), 2324–2337 (2015)

[43] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2014)

[44] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research (2014)

[45] Li, Q., Han, Z., Wu, X.-M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

[46] Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. Advances in neural information processing systems (2020)

Fenyu Hu received the B.S. degree in School of Electronical Engineering from Beijing University of Post and Communications, in 2017. He is currently pursuing Ph.D. degree in Center for Research on Intelligent Perception and Computing (CRIPAC) at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China.

His research interests include graph representation learning, machine learning, and recommender systems.

E-mail: fenyu.hu@cripac.ia.ac.cn

ORCID: 0000-0001-9881-0850

Dingshuo Chen received the B.S. degree in computer science from Shandong University, Qingdao, China, in 2022. He is currently pursuing a master's degree in Center for Research on Intelligent Perception and Computing (CRIPAC) at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China.

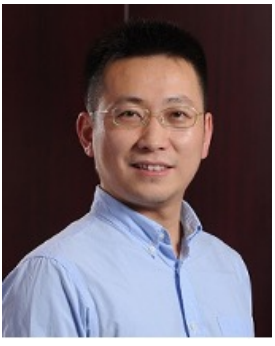
His research interests include machine learning, data mining, graph representation learning and AI for Science.

E-mail: dingshuo.chen@cripac.ia.ac.cn

Qiang Liu received his B.S. degree in electronic science from Yanshan University, China, in 2013, and Ph.D. degree from University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2018. He is an Associate Professor with Center for Research on Intelligent Perception and Computing (CRIPAC) at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA).

His research interests include machine learning, data mining, user modeling and information credibility evaluation.

E-mail: qiang.liu@nlpr.ia.ac.cn



Shu Wu received his B.S. degree from Hunan University, China, in 2004, M.S. degree from Xiamen University, China, in 2007, and Ph.D. degree from Department of Computer Science, University of Sherbrooke, Quebec, Canada, all in computer science. He is an Associate Professor with Center for Research on Intelligent Perception and Computing (CRIPAC) at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA).

His research interests include data mining, information retrieval, and recommendation systems.

E-mail: shu.wu@nlpr.ia.ac.cn

Response Letter

Dear Editor-in-chief, Associate Editor and Reviewers,

First of all, we would like to express our thanks to the reviewers for the time they have spent on our paper and for their valuable comments and constructive critiques. Their reviews are really helpful to improve our paper. Below, we first summarize the major changes between the revised manuscript and the original manuscript, and then address all the comments and questions raised by the reviewers one by one.

We have done a thorough revision of the original submission. The major changes can be summarized as follows:

(1) We added the pre-training details in Section 5.1.3.

(2) We added more experiments on Sider and Tox21 dataset in Section 5.3.

(3) We added the complexity analysis Section 4.4.

(4) We carefully corrected some grammar and presentation errors in the manuscript.

Sincerely yours,

Fenyu Hu, Dingshuo Chen, Qiang Liu, Shu Wu
Center for Research on Intelligent Perception and Computing (CRIPAC)
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences (CASIA)
Beijing, China

****

Editor Comments

Associate Editor
Comments to the Author:
The authors are recommended to revise the paper according to the reviewers' comments on paper writing, explanation of experimental results and settings, and more results on other datasets.

****

Response to Reviewer: 1

(1) In Section 5.1.2, the authors consider the setting of transfer learning. However, they do not provide clear details on how they pre-train the GNNs, such as the pre-training dataset, hyper-parameters, and other pre-training details. Clarifying these details would be helpful for readers who are interested in replicating the experiments or understanding the full scope of the proposed method.

Thank you very much for your comment. We have restated the training details for transfer learning in Section 5.1.3 as follows.

"For transfer learning, we closly follow the setting in \cite{xu2021self}. For fair comparison, we use the data sets as in \cite{hu2019strategies}. Specificically, we apply a subset of ZINC15 database\cite{sterling2015zinc}, which contains 2 million unlabeled molecules. For the pre-training strategy, we also use GraphLog\cite{xu2021self} as the base model. We adopt a five-layer GIN with 300-dimensional hidden units for all compared methods, including MMOE, GradNorm, VA-ST and IM-GNN. We use a linear classifier for fine-tuning and adopt an Adam optimizer(learning rate: 0.001). Unless otherwise specified, the batch size N is set as 512, and the hierarchical prototypes' depth L_p is set as 3."

(2) Grammar mistakes: "As a result, the proposed IM-GNN can be regard as a semi-supervised learning method".

Thanks for your carefulness. We have corrected this sentence: "As a result, the proposed IM-GNN can be regarded as a semi-supervised learning method". We also revise other sentences throughout the paper.

(3) The authors should provide more theoretical or experimental complexity analysis.

Thank you for pointing out this problem. We added the corresponding analysis in Section 4.4.

"Compared to vanilla multi-task GNNs, the additional computation complexity of IM-GNN comes from the pseudo-label generation process. Suppose there are m labels in the training set, then the bipartite graph contains $(|D|+M)$ nodes and m edges. Correspondingly, the complexity of the message propagation is $\mathcal{O}((|D|+M)md)$. In other words, the complexity is proportional to the size of training instances and training labels."

(4) More experiments on other datasets.

Thank you very much for your good suggestion. We added the imputation performance on Tox21 dataset. The results proves the effectiveness and the generalizability over different datasets of IM-GNN. The results are shown in Figure 3 and Figure 4.

(4) Notations throughout the paper.

Thanks for your suggestion. We summarized all the used notations in Table 1.

****

Reviewer: 2

(1) Some grammer and annotation mistakes need to be corrected.

Thank you for pointing out this problem. We apologize for the mistakes in the original manuscript. In the revised one, we have carefully corrected our presentation accordingly.

- In section 3.2, we restated the meaning of f_P^j.
- We removed the italic style in Section 4.
- In the first paragraph of Section 4.3, we revised the sentence as "pseudo-labels severely degrade the model performance."
- In section 4.3.2, we revised the sentence as "the proposed IM-GNN can be regarded as a semi-supervised learning method."
- In section 4.3.2, we revised the sentence as "We use a five-layer architecture."
- In section 5.1.3, "We use a five-laer architecture xxx" -> "We use a five-layer architecture for GCN and GIN."

(2) It is highly suggested that the author describe in detail the pre-training data set used in the transfer learning setting and what the pre-training strategy is, which is crucial for the comparison of performance.

Thanks for your suggestion. We have restated the training details for transfer learning in Section 5.1.3 as follows.

"For transfer learning, we closly follow the setting in \cite{xu2021self}. For fair comparison,

we use the data sets as in \cite{hu2019strategies}. Specificically, we apply a subset of ZINC15 database\cite{sterling2015zinc}, which contains 2 million unlabeled molecules. For the pre-training strategy, we also use GraphLog\cite{xu2021self} as the base model. We adopt a five-layer GIN with 300-dimensional hidden units for all compared methods, including MMOE, GradNorm, VA-ST and IM-GNN. We use a linear classifier for fine-tuning and adopt an Adam optimizer(learning rate: 0.001). Unless otherwise specified, the batch size N is set as 512, and the hierarchical prototypes' depth L_p is set as 3."