# Chemical Toxicity Prediction Based on Semi-supervised Learning and Graph Convolutional Neural Network

**Jiarui Chen**
University of Macau https://orcid.org/0000-0002-9681-6068
**Yain-Whar Si**
University of Macau Faculty of Science and Technology https://orcid.org/0000-0001-8468-6182
**Chon-Wai Un**
University of Macau Faculty of Science and Technology https://orcid.org/0000-0002-1555-9611
**Shirley W. I. Siu** ( shirleysiu@umac.mo )
University of Macau https://orcid.org/0000-0002-3695-7758

**Research Article**

**Keywords:** chemical toxicity, deep learning, graph convolutional neural network, semi-supervised learning, mean teacher, Tox21, ADMET

**Posted Date:** July 28th, 2021

**DOI:** https://doi.org/10.21203/rs.3.rs-733550/v1



**Version of Record:** A version of this preprint was published at Journal of Cheminformatics on November 27th, 2021. See the published version at https://doi.org/10.1186/s13321-021-00570-8.

METHODOLOGY

# Chemical Toxicity Prediction Based on Semi-supervised Learning and Graph Convolutional Neural Network

Jiarui Chen[1], Yain-Whar Si[1], Chon-Wai Un[1] and Shirley W. I. Siu[1,2*]

*Correspondence:
shirleysiu@um.edu.mo
[1]Department of Computer and Information Science, University of Macau, Avenida da Universidade, Taipa, 999078 Macau, China
Full list of author information is available at the end of the article

**Abstract**

As safety is one of the most important properties of drugs, chemical toxicology prediction has received increasing attentions in the drug discovery research. Traditionally, researchers rely on *in vitro* and *in vivo* experiments to test the toxicity of chemical compounds. However, not only are these experiments time consuming and costly, but experiments that involve animal testing are increasingly subject to ethical concerns. While traditional machine learning (ML) methods have been used in the field with some success, the limited availability of annotated toxicity data is the major hurdle for further improving model performance. Inspired by the success of semi-supervised learning (SSL) algorithms, we propose a Graph Convolution Neural Network (GCN) to predict chemical toxicity and trained the network by the Mean Teacher (MT) SSL algorithm. Using the Tox21 data, our optimal SSL-GCN models for predicting the twelve toxicological endpoints achieve an average ROC-AUC score of 0.757 in the test set, which is a 6% improvement over GCN models trained by supervised learning and conventional ML methods. Our SSL-GCN models also exhibit superior performance when compared to models constructed using the built-in DeepChem ML methods. This study demonstrates that SSL can increase the prediction power of models by learning from unannotated data. The optimal unannotated to annotated data ratio ranges between 1:1 and 4:1. This study demonstrates the success of SSL in chemical toxicity prediction; the same technique is expected to be beneficial to other chemical property prediction tasks by utilizing existing large chemical databases.

**Keywords:** chemical toxicity; deep learning; graph convolutional neural network; semi-supervised learning; mean teacher; Tox21; ADMET

## Introduction

The fundamental strategy in modern drug discovery and development is to identify chemical compounds that potently and selectively modulate the functions of the target molecules to elicit a desired biological response. How to quickly locate these compounds from the vast chemical space and then determine their drug-like properties remains a major challenge [1, 2, 3]. Traditionally, chemists and biologists perform *in vitro* and *in vivo* experiments to test the pharmacodynamics and pharmacokinetic (PD/PK) properties of selected candidates obtained from initial screening results [4, 5]. However, these experiments are not only very costly in terms of time and money, the experiments that involve animal testings are increasingly questionable from ethical perspectives [6]. Previous studies show that it typically

takes six to twelve years and more than 2.6 billion dollars to develop a new drug. Of this cost, about 1.1 billion dollars is for the drug development phases prior to human testing [7].

Toxicity is one of the five pharmacokinetic properties (ADMET) that must be strictly ascertained before a new drug candidate is approved for clinical trials [8]. On the premise that "*the structure of a chemical substance implicitly determines its physical and chemical properties and reactivity, and these properties interact with biological systems to determine its biological/toxicological properties*" [9, 10], efforts have been made to develop computational methods, often machine learning (ML) based, that attempt to relate the toxicological properties of compounds to their chemical structures. For a comprehensive review of ML-based toxicity prediction methods, the readers are referred to refs [11, 12, 13].

Graph Convolutional Neural Networks (GCN) are commonly used for tasks such as social network analysis and knowledge graph mining. Since biomolecular structures can also be represented as graphs, a variety of GCN-based biomolecular property prediction models have been developed in recent years. For example, the Weave model was proposed by Kearnes et al. in 2016 [14], which was a deep learning system based on molecular graph convolutions. This model uses only the simple descriptions of atoms, bonds, and atom pairs as input data. In addition, a learnable module called Weave module, extracts and combines the features of atom and distance relationship with learnable parameters. These modules can be stacked to an arbitrary depth to allow fine-tuning of the architecture for the needs of different learning tasks. In 2017, Li et al. proposed the GraphConv-SuperNode model [15]. By adding a dummy fully connected node (the super node) in each graph, this model captures and extracts graph-level representations from chemical structures, allowing it to focus on graph-level classification and regression tasks. In 2020, Wang et al. proposed a graph attention convolutional neural network (GACNN) that classified poisonous chemicals to honey bees [16], which is a Graph Convolution Neural Network with undirected graph and attention mechanism. They demonstrated that the performance of their GACNN model was better than all previous models, and they also summarised important structural features that might lead to poisoning.

All of these previous studies have highlighted the advantages of using GCN-based models to predict biomolecular properties. First, the suitability of different traditional molecular descriptors for different tasks significantly affects the performance of the models [17, 16]. Graph-based molecular representations can circumvent this problem by preserving the structural and physicochemical information of the molecules. Second, the majority of models using graph-based techniques perform better on biomolecular property prediction tasks than conventional ML models using traditional molecular descriptors[14, 15, 16, 18]. Third, since GCN-based models can directly manipulate graph-based molecular representations, they can retain molecular structural information during prediction. This characteristics of GCN makes the interpretability of GCN-based models superior to other traditional ML models.

Based on the different training strategies, ML algorithms can be broadly classified into 4 types, namely supervised learning (SL), semi-supervised learning (SSL), unsupervised learning and reinforcement learning [19]. All the prediction models we mentioned above are based on the SL algorithms which learn only from annotated datasets. However, despite enormous efforts in data curation and data sharing, the amount of labeled data falls far short of the amount of known compounds. Strategies to make use of the unannotated data such as those of SSL are expected to enhance the generalizability of prediction models.

Therefore, inspired by the success of GCN and the needs for improving chemical toxicity prediction confronted with limited data, we designed a learning system that hybridizes graph convolution neural network (GCN) and SSL to predict the toxicity of chemical compounds. Here, we used chemical data from the Tox21 dataset as annotated data and collected compounds from other datasets as unannotated data. First, the molecular features encoded in GCN were defined, then experiments were performed to investigate the influence of SSL on the predictivity of the models. Moreover, the performances of the SSL models with varying unannotated data ratios were compared, which showed that SSL has a positive influence on the prediction performance of GCN models.

This paper is organized as follows. The theoretical foundation of GCN and the mean teacher SSL algorithm are presented in the Material and Method section. The dataset, model, and validation technique are then described. The Results section contains comparative study of the traditional ML, SL-GCN, and SSL-GCN models performances. The impact of various unannotated data ratios was also investigated. Finally, SSL-GCN was compared to existing DeepChem methods for toxicity prediction.

## Material and Method

### Graph Convolutional Neural Network (GCN)

Traditional convolutional neural networks (CNN) can extract features from Euclidean or grid structure data, such as images and text. But for non-Euclidean data like social networks, knowledge graphs, or chemical structures, due to its irregular data topology, CNN cannot directly operate on them [20, 21]. A solution for machine learning on non-Euclidean data is Graph Convolutional Neural Network (GCN)[22]. GCN has been widely used in solving computer science problems such as social network analysis[23], natural language processing[24, 25], and recommendation system[26, 27], and also chemistry problems such as molecular properties prediction[18, 14, 20, 28]. For the latter, each molecule is described as an undirected graph where atoms are represented as nodes and covalent chemical bonds are represented as edges. The basic idea of graph convolution is to apply a learnable function on each node and its neighbors, gradually merging information from distant atoms through the connecting edges, and ultimately extracting the atom-type and connectivity patterns in the molecule. In this work, we used off-the-shelf GCN method that was proposed by Kipf et al. in 2017 [29]. The layer-wise propagation

function of this approach is defined in the following equations in terms of matrix calculation:

$$\tilde{A} = A + I \tag{1}$$

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{2}$$

These equations can be denoted as $f(H^{(l)}, A)$. $\tilde{A}$ represents the adjacency matrix $A$ of an undirected graph $\mathcal{G}$ with added self-connections $I$. $\tilde{D}$ is the degree matrix of $\tilde{A}$. $H^{(l)} \in \mathbb{R}^{N\times D}$ represents the nodes signal matrix (features) generated by the $l^{th}$ layer, where $N$ and $D$ denote the number of nodes in this graph and the dimension of each node's signal matrix respectively. $W^{(l)}$ is the layer-specific learnable weight matrix of the $l^{th}$ layer. $\sigma$ denotes a non-linear activation function [29].

To facilitate implementation, the previous equations can be represented as the following:

$$h_i^{(l+1)} = ReLU\left(b^{(l)} + \sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(i)|}\sqrt{|\mathcal{N}(j)|}} h_j^{(l)}W^{(l)}\right) \tag{3}$$

where $\mathcal{N}(i)$ is the set of neighbors of the node $i$. $W^{(l)}$ represents the layer-specific learnable weight matrix of the $l^{th}$ layer, $h_j^{(l)}$ is the signal matrix (features) of each neighbor node $j$ around $i$, and $b^{(l)}$ is the bias value of the $l^{th}$ layer. Therefore, the signal of each node in the next layer is determined by the weighted sum of signals in each node of the current layer and the signals of its adjacent nodes of the same layer. All signals are nonlinearly transformed using the Rectified Linear Unit (ReLU) function, $ReLU(x) = max(0, x)$.

### Semi-supervised Learning (SSL)

The basic idea of machine learning (ML) is to reproduce the human learning process by computer algorithms. Most ML algorithms can be classified into four types[30, 19]: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The most commonly used method is supervised learning. It derives knowledge from training data with fully annotated labels [31]. However, acquiring accurate annotated data is sometimes difficult for certain tasks such as chemical compound properties prediction. On one hand, there are tens of thousands known chemical compounds that exist in nature, and even more artificial chemical compounds are being produced every year. On the other hand, each annotation requires labor-intensive and expensive procedure from compound synthesis to measurement. Consequently, a significant amount of molecules are not properly labelled while some labels may subject to experimental errors. To learn from incompletely annotated data, semi-supervised learning method is more suitable [32].

In SSL, it is assumed that the label function is smooth in high-density areas, so data points located in the same area should share the same label. Based on this

smoothness assumption, even unlabelled data can be exploited in the learning process. Here, the main idea is to build classification models that are robust to local perturbations in the input data. When the input data is perturbed with a small amount of noise, the prediction results for the perturbed data and original data should be similar [33]. Since this consistency in predictions does not depend on the data labels, therefore unlabelled data can be exploited in the training process to enhance the prediction consistency of the model.

In this study, we implemented the SSL algorithm proposed by Tarvainen and Valpola, called Mean Teacher (MT) [34]. This algorithm requires two models with the same architecture, namely the student model and the teacher model. In each training epoch, the student model updates its internal weights based on the classification loss on the labeled data and the consistency loss between the two models on the unlabeled data. After the student model is updated, the teacher model is also updated using an exponential moving average (EMA) strategy [34, 32]. Previous studies have demonstrated that this kind of self-ensembling framework could bring improvements to classification models [34, 35]. The pseudo code of this algorithm is shown below:

**Algorithm 1:** Pseudo code of the Mean Teacher (MT) algorithm

**Data:** labeled training dataset $L$, unlabeled training dataset $U$

**repeat**
  **for** $x \in L + U$ **do**
    $z \leftarrow m_s(g(x), \theta_s^{i-1})$;
    $\tilde{z} \leftarrow m_t(g(x), \theta_t^{i-1})$;
    **if** $x \in L$ **then**
      $Loss_{cls} = CrossEntropy(z, label)$;
    **end**
    $Loss_{con} = MSE(z, \tilde{z})$;
    $Loss = w_i \times Loss_{con} + Loss_{cls}$;
    $\theta_s^i \leftarrow Update(Loss, \theta_s^{i-1})$;
    $\theta_t^i \leftarrow EMA(\theta_s^i, \theta_t^{i-1}, \alpha_i)$;
  **end**
**until** *end condition is met*;

$g(\cdot)$ denotes the data perturbation function, $m_s(\cdot)$ and $m_t(\cdot)$ represent the student and teacher models respectively, $\theta_s^i$ and $\theta_t^i$ represent the internal weights in the training step $i$, $z$ and $\tilde{z}$ are the generated classification probabilities. $Loss_{cls}$ and $Loss_{con}$ represent classification loss and consistency loss. $w_i$ denotes the consistency loss coefficient in the training step $i$. This consistency loss coefficient varies with the training steps. It is defined as the function $e^{-5(1-t)^2}$, where $t \in \{0, 1\}$, represents scaled number of training step [34]. $Update(\cdot)$ is the process of updating the internal weights of the model through backpropagation.

$EMA(\cdot)$ is the process of updating the weights in $m_t$ by applying the Exponential Moving Average (EMA) of weights in $m_s$ where $\alpha_i$ is the smoothing coefficient. The following equation defines this process mathematically:

$$\theta_t^i = \alpha_i \theta_t^{i-1} + (1 - \alpha_i)\theta_s^i \tag{4}$$

In our implementation, we applied the Gaussian noise $g(x)$ as the data perturbation method using the same distribution for both $m_s(\cdot)$ and $m_t(\cdot)$. The cross entropy loss function and Mean Squared Error (MSE) are used to compute the classification loss and consistency loss, respectively. The GCN network is optimized using the Adam optimizer [36], which is the optimizer chosen in the original implementation of MT [34]. Although both the well-trained teacher model and the student model can be used for prediction, previous studies have demonstrated that the teacher model is more accurate than the student model. [34, 32] Therefore, the teacher model is used as the final classification model.

**Figure 1 The SSL-GCN model for compound toxicity prediction.** Molecular compounds are converted into graphs of nodes and connections. The GCN model architecture is composed of two stacked layers of graph convolutional layer, dropout, and batch normalization layer. All signals are summarized by the max pooling layer and fed into the multilayer perceptron network to generate the final output. The teacher and student GCN models are updated using the MT algorithm.

## Datasets

For semi-supervised learning, both labeled (compounds with toxicity information) and unlabeled (compounds without toxicity information) data are required. In this study, the Tox21 dataset from MoleculeNet [37] is used as the labeled data. The Tox21 challenge is a community-wide compound toxicity prediction competition in 2014. Since then, the Tox21 dataset has been widely used as the benchmark dataset for evaluating toxicity prediction models. It consists of 12 endpoints, including 7 nuclear receptor signals (NR-AR, NR-AhR, NR-AR-LBD, NR-ER, NR-ER-LBD, NR-Aromatase, NR-PPAR-gamma) and 5 stress response indicators (SR-ARE, SR-ATAD5, SR-HSE, SR-MMP, SR-p53). In this dataset, each compound is expressed in Simplified Molecular Input Line Entry Specification (SMILES) format and the binary labels indicate whether the compound is toxic to a specific toxicological endpoint. In total, the Tox21 dataset include 7831 compounds and 12 different endpoints. It should be noted that not all compounds have all endpoint labels; the missing endpoint label means that the toxicology effect toward this endpoint is unknown. For unlabeled data, other chemical compound datasets were sought from the MoleculeNet website, including ClinTox, SIDER, ToxCast, and HIV datasets [37]. All the label information in these datasets have been removed. In addition, duplicate molecules between these datasets and the Tox21 dataset have also been removed. In total, 50527 compounds were used as unlabeled data. Table 1 shows the details of the datasets used in this study.

For each labeled dataset, we follow the conventional dataset splitting rule with the splitting ratios of 0.8:0.1:0.1 to divide the dataset into training, validation and test

**Table 1** The labeled compound toxicity datasets for 12 toxicological endpoints and the unlabeled dataset.

| Endpoint | Compounds(Labeled) | Training Set | Validation Set | Test set |
|---|---|---|---|---|
| NR-AhR | 6549 | 5239 | 655 | 655 |
| NR-AR-LBD | 6758 | 5406 | 676 | 676 |
| NR-AR | 7265 | 5812 | 726 | 727 |
| NR-Aromatase | 5821 | 4656 | 582 | 583 |
| NR-ER-LBD | 6955 | 5564 | 695 | 696 |
| NR-ER | 6193 | 4954 | 619 | 620 |
| NR-PPAR-gamma | 6450 | 5160 | 645 | 645 |
| SR-ARE | 5832 | 4665 | 583 | 584 |
| SR-ATAD5 | 7072 | 5657 | 707 | 708 |
| SR-HSE | 6467 | 5173 | 647 | 647 |
| SR-MMP | 5810 | 4648 | 581 | 581 |
| SR-p53 | 6774 | 5419 | 677 | 678 |
| Unlabeled data | 50527 | — | — | — |

sets. Training set is used for the training process, validation set for the hyperparameter tuning process and the test set is to measure the generalization performance. The most commonly used splitting method is random splitting. However, it is not always suitable for molecular data because random splitting cannot guarantee that the training and test sets contain diverse and representative data samples [37, 38]. In order to overcome the problem of data bias, we adopted a scaffold splitting method. It splits the dataset according to the two-dimensional structural framework of the molecule[39, 40] and then assign structurally different molecules into different subsets [37]. In this way, both the training set and the test set contain a good proportion of data samples scattered in the molecular space of the dataset, and we can expect that the performance of the model measured on this test set is closer to its actual performance on new data.

As mentioned above, an undirected graph can be described by two matrices, namely the signal (feature) matrix $H$ and the adjacency matrix $A$. In this study, we used the molecule-graph conversion tool from Deep Graph Library (DGL) [41] to convert molecules from SMILES to graphs. For each molecule, the connectivity of atoms is stored in the adjacency matrix and the physicochemical properties of each atom (node features) are encoded into a feature matrix in binary or numerical form. Since the DGL conversion tool provides eight default atom features, as listed in Table 2, the dimension of each node feature matrix is $1 \times 74$. Therefore, for a molecule with $N$ atoms, the conversion will generate one adjacency matrix of dimension $N \times N$ and one feature matrix of dimension $N \times 74$. This graph conversion process is depicted in Figure 1. After this step, the graph-based molecular data can be learned by the graph convolutional neural network.

**Table 2** Atom features provided by the molecule-graph conversion tool from Deep Graph Library.

| No. | Description | No. of bits | Form |
|---|---|---|---|
| 1 | One hot encoding of the atom type | 1-43 | Binary |
| 2 | One hot encoding of the atom degree | 44-54 | Binary |
| 3 | One hot encoding of the number of implicit Hs on the atom | 55-61 | Binary |
| 4 | Formal charge of the atom | 62 | Numerical |
| 5 | Number of radical electrons of the atom | 63 | Numerical |
| 6 | One hot encoding of the atom hybridization | 64-68 | Binary |
| 7 | Whether the atom is aromatic | 69 | Numerical |
| 8 | One hot encoding of the number of total Hs on the atom | 70-74 | Binary |

### Model Architecture and Hyperparameters Selection

The architecture of our GCN model consists of two parts, an encoder and a classifier. The encoder extracts and updates node representations through several graph convolutional layers (Graph Conv). In addition, there is a dropout layer after each Graph Conv layer to provide additional noise to the molecular representations.[34, 32] The last layer of the encoder merges all nodes features into a tensor by using max-pooling and weighted sum operations. This tensor is the learned representation of the input molecule. The classifier is to compute the final prediction. We used the classifier provided in DGL [41] which contains two layers perceptron (MLP) with a dropout layer and a batch normalization layer.

In order to select the best hyperparameters for these models, Bayesian optimization algorithm [42] is used to search the hyperparameter space, and the maximum number of trials is 32. In each trial, the algorithm selects a set of candidate hyperparameters and initializes the model. Then, model training and validation are carried out iteratively until the early stopping condition of 30 epochs is met. After all trials are completed, a set of candidate hyperparameters with the best validation metric (ROC-AUC) is selected as the default hyperparameters for the following experiments.

Since the toxicity dataset is highly imbalanced, with an average toxic/non-toxic data ratio of about 1:17, the area under the Receiver Characteristic Operator curve (ROC-AUC) is used as the main metric in the hyperparameter selection process (practically, to decide for early stopping) and the final model evaluation. The hyperparameters with the best validation performance are selected to construct the optimal toxicity prediction models. Finally, the generalization performance of these models are estimated using the test set.

### Implementation Detail

In this study, all implementations and experiments are carried out in an environment with following libraries/software: Python 3.7.9, Anaconda 4.7.10, Scikit-learn 0.23.2, RDKit v2018.09.3.0. We used Pytorch 1.7.0 with CUDA 10.0 as the basic machine learning framework. The GCN model is implemented using DGL 0.5.6 and its supplementary package DGL-LifeSci 0.2.6 [41] (available on GitHub, DGL [43], DGL-LifeSci [44]). The Bayesian Optimization process for hyperparameter selection is implemented using Hyperopt 0.2.5 [42] (available on GitHub [45]). We also used DeepChem 2.5.0 [46] to generate the benchmark scores of other state-of-the-art models on the Tox21 dataset (available on GitHub [47]).

## Results

All experiments were repeated five times to observe the variability of the results and obtain an accurate measure of model performance through the average ROC-AUC score. The complete record of all experiments can be found in the supplementary information.

### Performance of Conventional Machine Learning (ML) Methods

To establish the baseline performance, several commonly used ML algorithms, namely K-Nearest Neighbor (KNN), Neural Network (NN), Random Forest (RF), Support Vector Machine (SVM) and eXtreme Gradient Boosting (XGBoost) were tested. The compounds were encoded using the Extended Connectivity Fingerprints (ECFP4), which is a circular topological fingerprint designed for molecular characterization, similarity searching, and structure-activity modeling [48]. The encoding was generated using the RDKit library. In total, 60 different ML models (12 prediction tasks × 5 types of ML algorithms) were trained and optimized using the training and validation sets. Subsequently, the optimal models were tested on the test set. The test performance of these conventional models on the 12 toxicity prediction tasks are presented in Table 3. Each experiment was repeated 5 times; the average ROC-AUC score and the standard deviation (std) were reported. In all prediction tasks, the ROC-AUC scores range between 0.5127 and 0.8287. In certain cases (KNN, SVM, and XGBoost), we observed that the same optimal models were obtained in all replicate experiments such that the ROC-AUC scores are the same (std = 0). Overall, RF, XGBoost, and SVM generated the best models for 5, 4, 3 of the prediction tasks, respectively. The average ROC-AUC score of the best performing ML models of all tasks is 0.71.

**Table 3** The average test performance of conventional ML models on the 12 prediction tasks in 5 repeated experiments.

| Tasks | KNN | | NN | | RF | | SVM | | XGBoost | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | Std | AUC | Std | AUC | Std | AUC | Std | AUC | Std |
| NR-AR-LBD | 0.6955 | - | 0.6671 | 0.0244 | **0.7323** | 0.0267 | 0.6795 | - | 0.6784 | - |
| NR-AR | 0.6527 | - | 0.6806 | 0.0088 | 0.6836 | 0.0266 | **0.7193** | - | 0.6818 | - |
| NR-AhR | 0.7639 | - | 0.7628 | 0.0177 | 0.8243 | 0.0074 | 0.7794 | - | **0.8287** | - |
| NR-Aromatase | 0.5576 | - | 0.5127 | 0.0772 | 0.6900 | 0.0092 | 0.6873 | - | **0.7106** | - |
| NR-ER-LBD | 0.6191 | - | 0.5387 | 0.1171 | 0.6169 | 0.0300 | 0.6078 | - | **0.6250** | - |
| NR-ER | 0.6597 | - | 0.6549 | 0.0162 | 0.6316 | 0.0080 | 0.6126 | - | **0.6745** | - |
| NR-PPAR-gamma | 0.6182 | - | 0.5558 | 0.0736 | **0.7135** | 0.0258 | 0.6454 | - | 0.6414 | - |
| SR-ARE | 0.6366 | - | 0.5656 | 0.0251 | 0.6603 | 0.0018 | **0.6843** | - | 0.6640 | - |
| SR-ATAD5 | 0.5866 | - | 0.6240 | 0.0537 | **0.6928** | 0.0189 | 0.6546 | - | 0.6841 | - |
| SR-HSE | 0.6574 | - | 0.6143 | 0.0222 | 0.6852 | 0.0131 | **0.6858** | - | 0.6647 | - |
| SR-MMP | 0.7057 | - | 0.6551 | 0.0612 | **0.7818** | 0.0065 | 0.7794 | - | 0.7656 | - |
| SR-p53 | 0.6778 | - | 0.5963 | 0.0075 | **0.7263** | 0.0130 | 0.7051 | - | 0.6942 | - |

### Performance of Supervised Learning GCN (SL-GCN)

Having established the baseline performance of the traditional ML models in toxicity prediction, we went on to test the GCN models for the 12 prediction tasks. Similar to other ML models above, the GCN models were trained using supervised learning and optimized by the Bayesian optimization algorithm, hence the name SL-GCN. In Figure 2, the ROC curves of the SL-GCN models on the test set prediction are plotted against other ML models, and the 5-repeated average of the ROC-AUC scores are tabulated in Table 4. The results show that, while the SL-GCN models perform similarly to the best conventional ML models in the majority of the twelve toxicity prediction tasks, they improve in four of the tasks, including NR-ER, SR-ARE, SR-HSE, and SR-MMP, while they perform worse in three of the tasks, including NR-AR-LBD, NR-PPAR-gamma, SR-p53.

**Figure 2 ROC curves of conventional ML models and SL-GCN models.** The comparison of ROC curves between conventional ML models (black line) and SL-GCN models (red line) on 12 toxicity prediction tasks. Additional information of the ROC curves are provided in the supplementary information.

**Table 4** The average test performance of SSL-GCN models with various unlabeled data ratio ($R_u$ in brackets) on the 12 prediction tasks in 5 repeated experiments. For comparison, the results of the SL-GCN models are shown.

| Tasks | SL-GCN | | SSL-GCN (0.5) | | SSL-GCN (1.0) | |
|---|---|---|---|---|---|---|
| | AUC | Std | AUC | Std | AUC | Std |
| NR-AR-LBD | 0.6783 | 0.0269 | 0.7417 | 0.0105 | 0.7333 | 0.0401 |
| NR-AR | 0.7157 | 0.0367 | 0.7550 | 0.0483 | 0.7858 | 0.0357 |
| NR-AhR | 0.8260 | 0.0055 | 0.8161 | 0.0121 | 0.8295 | 0.0129 |
| NR-Aromatase | 0.7092 | 0.0167 | 0.7202 | 0.0057 | 0.7306 | 0.0156 |
| NR-ER-LBD | 0.6340 | 0.0161 | 0.6623 | 0.0330 | 0.6794 | 0.0411 |
| NR-ER | 0.6899 | 0.0160 | **0.7188** | 0.0196 | 0.7114 | 0.0179 |
| NR-PPAR-gamma | 0.6753 | 0.0278 | 0.7267 | 0.0210 | **0.7614** | 0.0212 |
| SR-ARE | 0.7134 | 0.0137 | 0.7241 | 0.0065 | 0.7288 | 0.0063 |
| SR-ATAD5 | 0.6850 | 0.0223 | 0.7119 | 0.0080 | 0.7061 | 0.0245 |
| SR-HSE | 0.7644 | 0.0096 | 0.7636 | 0.0239 | 0.7678 | 0.0080 |
| SR-MMP | 0.7988 | 0.0066 | **0.8120** | 0.0075 | 0.8035 | 0.0061 |
| SR-p53 | 0.6970 | 0.0253 | 0.7291 | 0.0114 | 0.7401 | 0.0203 |

| Tasks | SSL-GCN (2.0) | | SSL-GCN (3.0) | | SSL-GCN (4.0) | |
|---|---|---|---|---|---|---|
| | AUC | Std | AUC | Std | AUC | Std |
| NR-AR-LBD | **0.7647** | 0.0279 | 0.7377 | 0.0145 | 0.7477 | 0.0135 |
| NR-AR | 0.7512 | 0.0358 | 0.7412 | 0.0659 | **0.7967** | 0.0251 |
| NR-AhR | 0.8287 | 0.0072 | **0.8303** | 0.0055 | 0.8224 | 0.0090 |
| NR-Aromatase | 0.7232 | 0.0040 | 0.7287 | 0.0082 | **0.7337** | 0.0057 |
| NR-ER-LBD | 0.6772 | 0.0161 | 0.6662 | 0.0250 | **0.6870** | 0.0282 |
| NR-ER | 0.7039 | 0.0124 | 0.7113 | 0.0083 | 0.7166 | 0.0137 |
| NR-PPAR-gamma | 0.7491 | 0.0201 | 0.7429 | 0.0177 | 0.7456 | 0.0223 |
| SR-ARE | **0.7297** | 0.0080 | 0.7277 | 0.0067 | 0.7243 | 0.0114 |
| SR-ATAD5 | 0.7096 | 0.0139 | **0.7175** | 0.0143 | 0.7077 | 0.0162 |
| SR-HSE | **0.7822** | 0.0097 | 0.7731 | 0.0098 | 0.7700 | 0.0066 |
| SR-MMP | 0.8100 | 0.0033 | 0.8031 | 0.0088 | 0.8081 | 0.0078 |
| SR-p53 | **0.7518** | 0.0198 | 0.7359 | 0.0147 | 0.7434 | 0.0126 |

## Performance of Semi-Supervised Learning GCN (SSL-GCN)

The MT technique employed in this study necessitates the use of two models with the same architecture, one for $m_t$ and one for $m_s$. Therefore, we used the hyper-parameters obtained from the SL-GCN models as the initial parameters to train SSL-GCN. As shown in the previous study [34], the amount of unlabeled data in the training process can affect the final model performance. To investigate this impact on the performance of the SSL-GCN models, we ran numerous trials with varying amounts of unlabeled data. We define the unlabeled-to-labeled data ratio as $R_u \in \{0.5, 1.0, 2.0, 3.0, 4.0\}$. So, when $R_u = 0.5$, the amount of unlabeled data is only half of that of the labeled data. Due to significant increase in training time, a large $R_u$, such as $> 4.0$, were not considered. Table 4 shows the test results of the optimized SSL-GCN models for the 12 toxicity prediction tasks, as well as a comparison of the ROC curves in Figure 3.

**Figure 3 ROC curves of best SSL-GCN, SL-GCN, and CM models.** The comparison of ROC curves between the best conventional ML models (CM, black line), SL-GCN models (blue line), and SSL-GCN models with the best $R_u$ (red line) on 12 toxicity prediction tasks. Additional information on the ROC curves can be found in the supplementary information.

**Figure 4 Comparison of AUC scores between SL-GCN, SSL-GCN and CM models** Comparison of the best models from conventional methods (CM), SL-GCN, and the SSL-GCN on twelve toxicity prediction tasks. The mean and standard deviation are obtained from the 5-repeat experiments.

As shown in Table 4, SSL improves the predictive power of the GCN models when sufficient amount of unlabeled data is included in the training. SSL-GCN with $R_u$ of 0.5 improves the ROC-AUC score in 10 of the 12 prediction tasks, while only the ROC-AUC scores of two tasks are somewhat reduced. When the SSL-GCN models are trained with additional unlabeled data ($R_u$= 1.0 to 4.0) , they always outperform their SL-GCN counterparts in terms of prediction accuracy. Nonetheless, the best $R_u$ for each prediction task is different. SSL-GCN produces 4 optimal models when $R_u = 2.0$; 3 optimal models when $R_u = 4.0$; 2 optimal models when $R_u = 0.5$, and 1 optimal model when $R_u = 1.0$. As a result, the best $R_u$ varies depending on the prediction task at hand. The rates of performance improvement in terms of ROC-AUC for different task range from 1% to 13%. Finally, Figure 4 compares the best CM, SL-GCN and SSL-GCN models. As can be clearly seen, SSL-GCN can produce models with greater predictive potential than CM and SL-GCN in all toxicity prediction tasks.

As a summary, the comparative study of the SSL-GCN models with varying $R_u$ values suggests that when training with unlabeled data, the ratio of unlabeled and labeled data should be treated as a hyperparameter in order to obtain the optimal model.

### Performance Comparison of SSL-GCN to the Built-In DeepChem Methods

The DeepChem package [46] provides some built-in ML methods that can be readily used to generate predictive models for different computational chemistry challenges. Making use of the DeepChem-integrated MoleculeNet datasets [37], we performed experiments to evaluate the performances of the DeepChem models on the Tox21 dataset. The dataset was splitted by scaffold splitting method and all models were initialized with the hyperparameters provided by the DeepChem package. Following the previous experimental procedure, we conducted the training, validation and test processes, and repeated them five times for each model. Here, we benchmark our method by comparing the performance of the SL-GCN and SSL-GCN models in the test set to these DeepChem models in terms of the average ROC-AUC score.

As shown in Table 5, among the 8 DeepChem models, the best one is *kernelsvm*, with an overall score of 0.7, whereas both our models SL-GCN and SSL-GCN beat the best DeepChem model with overall scores of 0.7156 (2% improvement) and 0.7571 (8% improvement), respectively. It should be mentioned that while the *graphconv* model utilizes similar graph convolution technique to our method but its use of different model architecture and molecular feature rendering their model less effective.

## Discussion and Conclusions

In this work, we attempt to improve compound toxicity prediction using graph convolutional neural network (GCN) and semi-supervised learning (SSL). We choose

**Table 5** Comparison of our GCN models (SL-GCN and SSL-GCN) and the models constructed using the DeepChem built-in ML methods. The overall score is the average ROC-AUC score in predicting the 12 prediction tasks in the test set. The experiments were repeated 5 times.

| Model | Description | Overall Score | Std. | Ref. |
|---|---|---|---|---|
| logreg | logistic regression model | 0.6397 | - | [49] |
| tf | deep neural network | 0.6582 | 0.0097 | [37] |
| tf-robust | deep neural network (with bypass layers) | 0.6825 | 0.0056 | [50] |
| rf | random forest model | 0.6618 | 0.0066 | [49] |
| kernelsvm | kernel SVM model | 0.7000 | - | [49] |
| graphconv | graph convolutional model | 0.6943 | 0.0043 | [51] |
| irv | Influence relevance voting (IRV) classifier | 0.6853 | - | [52] |
| xgb | xgboost classification model | 0.6908 | 0.0039 | [53] |
| SL-GCN | supervised GCN model | 0.7156 | 0.0068 | this study |
| SSL-GCN | semi-supervised GCN model | **0.7571** | 0.0084 | this study |

Mean Teacher [34] as the SSL algorithm to improve the prediction performance of GCN on 12 toxicity prediction tasks from the Tox21 dataset. Meanwhile, we hope to answer two questions about predictive modeling in this research. First, is GCN superior to other more commonly used ML methods? Second, is unlabeled data advantageous for model training?

To this end, we have designed and implemented a GCN model for chemical compounds based on simple physicochemical properties of atoms. Unlike other commonly used chemical fingerprints that represent an entire compound in a one-dimensional feature vector for learning, GCN encodes it into a network of features, where the network resembles bond connectivity in the molecule. Given that structural diversity of a dataset is one of the elements that affect the prediction performance and generalizability of a model, we have used the scaffold splitting approach to divide the dataset into training, validation, and test sets for each prediction task. The Bayesian optimization technique has been used to speed up the process of tuning hyperparameters.

Now, with the GCN model in place, we have trained and optimized the supervised learning SL-GCN models and the semi-supervised learning SSL-GCN models on 12 toxicity prediction tasks. To answer the first question, is GCN superior to other commonly used ML methods? We have trained and optimized toxicity prediction models using 5 conventional ML methods in the supervised learning setting. Our comparative study has revealed that out of the 12 prediction tasks, 5 tasks are better predicted by SL-GCN, 2 tasks are similarly predicted, and 5 tasks are worse by SL-GCN; and the "better" models are not improved by a large margin. Therefore, our experimental result suggests that in the same supervised learning setting, GCN is not superior to conventional ML methods. The answer to this question is a bit disappointing though, as a GCN model is much more complex and expensive to train than the conventional models.

We believe that the bottleneck to improvement is the limitation of available data. Instead of adding more annotated data, which is not always possible or easy, we turn our attention to unlabeled data. Here, we have applied the SSL algorithm, called Mean Teacher (MT), to enhance the performance of the GCN model. Encouragingly, SSL-GCN models consistently outperform their SL-GCN counterparts, with

the ROC-AUC scores improving between 1% and 13%. Nonetheless, the amount of unlabeled data required to boost performance has to be determined on a case-by-case basis. We have found that for the prediction of various toxicological endpoints, the appropriate ratios of unlabel-to-label data range from 1 to 4. Larger ratios may improve further, but were not investigated in this study due to limited computational resources. Finally, a comparative analysis of our models with the models from the DeepChem library was done. The findings are that the SL-GCN models are 2% to 12% better than the DeepChem models in terms of ROC-AUC, while the SSL-GCN models are 8% to 18% better. Based on the above results, our answer to the second question, "Is unlabeled data advantageous for model training?", is therefore yes, and the amount of unlabeled required to optimize the model is subject to each study.

In many bioinformatics tasks, the size of an annotated dataset is often limited, which complicates the implementation and limits the performance of many ML algorithms. The result of this study suggests that SSL could be applied to other property prediction tasks such as adsorption/distribution/metabolism/excretion (ADME), solubility, binding activity, etc., to improve the predictive ability of model by using unannotated data.

This study does, however, have some limitations. First, the toxicity of a compound is determined by several factors such as chirality and the nature of functional groups. This information requires a more delicate coding approach to avoid information loss during graph conversion. Although there are various well-designed molecular fingerprints or descriptors for conventional ML algorithms that can be used, there is no specific one that is suitable for GCN. Therefore, we have to use the molecule-graph conversion tool from Deep Graph Library (DGL) to convert molecules from SMILES to graphs. However, the graphs converted by this tool only include few basic molecular physicochemical properties. Due to the limited computational power, the running time of the graph convolution layers using the current feature matrix was already very high and adding additional features will certainly cost more time during the model development process. In our future study, it becomes particularly important to increase the diversity of molecular information contained in the feature matrix while limiting the size of the matrix. Second, the interpretability of our graph convolution model has not been explored. Most researchers consider ML methods with neural networks as a black box. The only factor that can be confirmed during the training or prediction process is the input data, and the prediction results produced by these ML models are unexplainable. Specifically for biomedical ML applications, this limitation has been amplified. Without knowing which part of the compound led to the prediction result, researchers cannot modify the original compounds or select the compounds with better structure to conduct further studies. Therefore, in the next step of our study, we will focus on the interpretability of the neural graph convolutional network. Finally, our study has exploited the SSL algorithm that is based on the self-ensembling framework. There are other recently proposed SSL algorithms, such as Mixup [54], Interpolation Consistency Training [55], ReMixMatch [56], FixMatch [57], etc. The impact of different SSL algorithms on the toxicity prediction needs further research.

**Availability of data and materials**
All data used in this study comes from MoleculeNet (`http://moleculenet.ai/`). The data and script files for reproducing the experiments can be downloaded from `https://github.com/chen709847237/SSL-GCN`. The final prediction models we trained in this study are also available.

**Competing interests**
The authors declare that they have no competing interests.

**Funding**
This project was supported by University of Macau (grant no. MYRG2019-00098-FST).

**Author's contributions**
S.W.I.S. and Y.W.S. conceived the study, J.C. designed the solution, conducted the experiments, analyzed the results, and drafted the manuscript. C.W.U. implemented some of the programs. S.W.I.S. and Y.W.S. finalized the manuscript. All the authors read and approved the final manuscript.

**Declarations**
We thank the Faculty of Science and Technology at University of Macau for providing the needed computing facilities.

**Author details**
[1]Department of Computer and Information Science, University of Macau, Avenida da Universidade, Taipa, 999078 Macau, China. [2] (present address) Institute of Science and Environment, University of Saint Joseph, Rua de Londres 106, 999078 Macau, China. [3]School of Pharmaceutical Sciences, Universiti Sains Malaysia, USM, 11800 Penang, Malaysia.

**References**
1. Llanos, E.J., Leal, W., Luu, D.H., Jost, J., Stadler, P.F., Restrepo, G.: Exploration of the chemical space and its three historical regimes. Proceedings of the National Academy of Sciences **116**(26), 12660–12665 (2019)
2. McInnes, C.: Virtual screening strategies in drug discovery. Current opinion in chemical biology **11**(5), 494–502 (2007)
3. Kubinyi, H., Mannhold, R., Timmerman, H.: Virtual Screening for Bioactive Molecules vol. 10. John Wiley & Sons, Weinheim, Germany (2008)
4. Dean, A., Lewis, S.: Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics. Springer, Berlin, Germany (2006)
5. Oprea, T.I., Matter, H.: Integrating virtual screening in lead discovery. Current opinion in chemical biology **8**(4), 349–358 (2004)
6. Bailey, J., Balls, M.: Recent efforts to elucidate the scientific validity of animal-based drug tests by the pharmaceutical industry, pro-testing lobby groups, and animal welfare organisations. BMC Med Ethics **20**, 16 (2019)
7. Pu, L., Naderi, M., Liu, T., Wu, H.-C., Mukhopadhyay, S., Brylinski, M.: e toxpred: a machine learning-based approach to estimate the toxicity of drug candidates. BMC Pharmacology and Toxicology **20**(1), 2 (2019)
8. Raies, A.B., Bajic, V.B.: In silico toxicology: computational methods for the prediction of chemical toxicity. Wiley Interdisciplinary Reviews: Computational Molecular Science **6**(2), 147–172 (2016)
9. McKinney, J.D., Richard, A., Waller, C., Newman, M.C., Gerberick, F.: The Practice of Structure Activity Relationships (SAR) in Toxicology. Toxicological Sciences **56**(1), 8–17 (2000)
10. Roy, K., Kar, S., Das, R.: Chapter 7—validation of qsar models. Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment; Roy, K., Kar, S., Das, RN, Eds, 231–289 (2015)
11. Wu, Y., Wang, G.: Machine learning based toxicity prediction: from chemical structural description to transcriptome analysis. International journal of molecular sciences **19**(8), 2358 (2018)
12. Idakwo, G., Luttrell, J., Chen, M., Hong, H., Zhou, Z., Gong, P., Zhang, C.: A review on machine learning methods for in silico toxicity prediction. Journal of Environmental Science and Health, Part C **36**(4), 169–191 (2018)
13. Yang, H., Sun, L., Li, W., Liu, G., Tang, Y.: In silico prediction of chemical toxicity for drug design using machine learning methods and structural alerts. Frontiers in Chemistry **6**, 30 (2018). doi:10.3389/fchem.2018.00030
14. Kearnes, S., McCloskey, K., Berndl, M., Pande, V., Riley, P.: Molecular graph convolutions: moving beyond fingerprints. Journal of computer-aided molecular design **30**(8), 595–608 (2016)
15. Li, J., Cai, D., He, X.: Learning graph-level representation for drug discovery. arXiv preprint arXiv:1709.03741 (2017)
16. Wang, F., Yang, J.-F., Wang, M.-Y., Jia, C.-Y., Shi, X.-X., Hao, G.-F., Yang, G.-F.: Graph attention convolutional neural network model for chemical poisoning of honey bees' prediction. Science Bulletin **65**(14), 1184–1191 (2020)
17. Lusci, A., Pollastri, G., Baldi, P.: Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. Journal of chemical information and modeling **53**(7), 1563–1575 (2013)
18. Feinberg, E.N., Sur, D., Wu, Z., Husic, B.E., Mai, H., Li, Y., Sun, S., Yang, J., Ramsundar, B., Pande, V.S.: Potentialnet for molecular property prediction. ACS central science **4**(11), 1520–1530 (2018)
19. Portugal, I., Alencar, P., Cowan, D.: The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications **97**, 205–227 (2018)
20. Altae-Tran, H., Ramsundar, B., Pappu, A.S., Pande, V.: Low data drug discovery with one-shot learning. ACS central science **3**(4), 283–293 (2017)

21. Rao, B., Zhang, L., Zhang, G.: Acp-gcn: The identification of anticancer peptides based on graph convolution networks. IEEE Access **8**, 176005–176011 (2020)
22. Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: Can gcns go as deep as cnns? In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9267–9276 (2019)
23. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 817–826 (2009)
24. Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. arXiv preprint arXiv:1703.04826 (2017)
25. Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., Sima'an, K.: Graph convolutional encoders for syntax-aware neural machine translation. arXiv preprint arXiv:1704.04675 (2017)
26. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)
27. Monti, F., Bronstein, M.M., Bresson, X.: Geometric matrix completion with recurrent multi-graph neural networks. arXiv preprint arXiv:1704.06803 (2017)
28. Li, J., Cai, D., He, X.: Learning graph-level representation for drug discovery. arXiv preprint arXiv:1709.03741 (2017)
29. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
30. Chen, J., Siu, S.W.: Machine learning approaches for quality assessment of protein structures. Biomolecules **10**(4), 626 (2020)
31. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering **160**, 3–24 (2007)
32. Cui, W., Liu, Y., Li, Y., Guo, M., Li, Y., Li, X., Wang, T., Zeng, X., Ye, C.: Semi-supervised brain lesion segmentation with an adapted mean teacher model. In: International Conference on Information Processing in Medical Imaging, pp. 554–565 (2019). Springer
33. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (2020)
34. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. arXiv preprint arXiv:1703.01780 (2017)
35. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
36. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
37. 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 **9**(2), 513–530 (2018)
38. Sheridan, R.P.: Time-split cross-validation as a method for estimating the goodness of prospective prediction. Journal of chemical information and modeling **53**(4), 783–790 (2013)
39. Bemis, G.W., Murcko, M.A.: The properties of known drugs. 1. molecular frameworks. Journal of medicinal chemistry **39**(15), 2887–2893 (1996)
40. RDKit: Open-Source Cheminformatics Software (2006). `https://www.rdkit.org/` Accessed 14 Jul 2021
41. Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., et al.: Deep graph library: Towards efficient and scalable deep learning on graphs. (2019)
42. Bergstra, J., Yamins, D., Cox, D.: Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: International Conference on Machine Learning, pp. 115–123 (2013). PMLR
43. DGL: Deep Graph Library (2018). `https://github.com/dmlc/dgl` Accessed 14 Jul 2021
44. DGL-LifeSci (2020). `https://github.com/awslabs/dgl-lifesci` Accessed 14 Jul 2021
45. Hyperopt: Distributed Hyperparameter Optimization (2018). `https://github.com/hyperopt/hyperopt` Accessed 14 Jul 2021
46. Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., Wu, Z.: Deep Learning for the Life Sciences. O'Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA (2019)
47. DeepChem (2015). `https://github.com/deepchem/deepchem` Accessed 14 Jul 2021
48. Rogers, D., Hahn, M.: Extended-connectivity fingerprints. Journal of chemical information and modeling **50**(5), 742–754 (2010)
49. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
50. Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R.P., Pande, V.: Is multitask deep learning practical for pharma? Journal of chemical information and modeling **57**(8), 2068–2076 (2017)
51. Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P.: Convolutional networks on graphs for learning molecular fingerprints. arXiv preprint arXiv:1509.09292 (2015)
52. Swamidass, S.J., Azencott, C.-A., Lin, T.-W., Gramajo, H., Tsai, S.-C., Baldi, P.: Influence relevance voting: an accurate and interpretable virtual high throughput screening method. Journal of chemical information and modeling **49**(4), 756–766 (2009)
53. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
54. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
55. Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825 (2019)
56. Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint

arXiv:1911.09785 (2019)
57. Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020)

# Figures

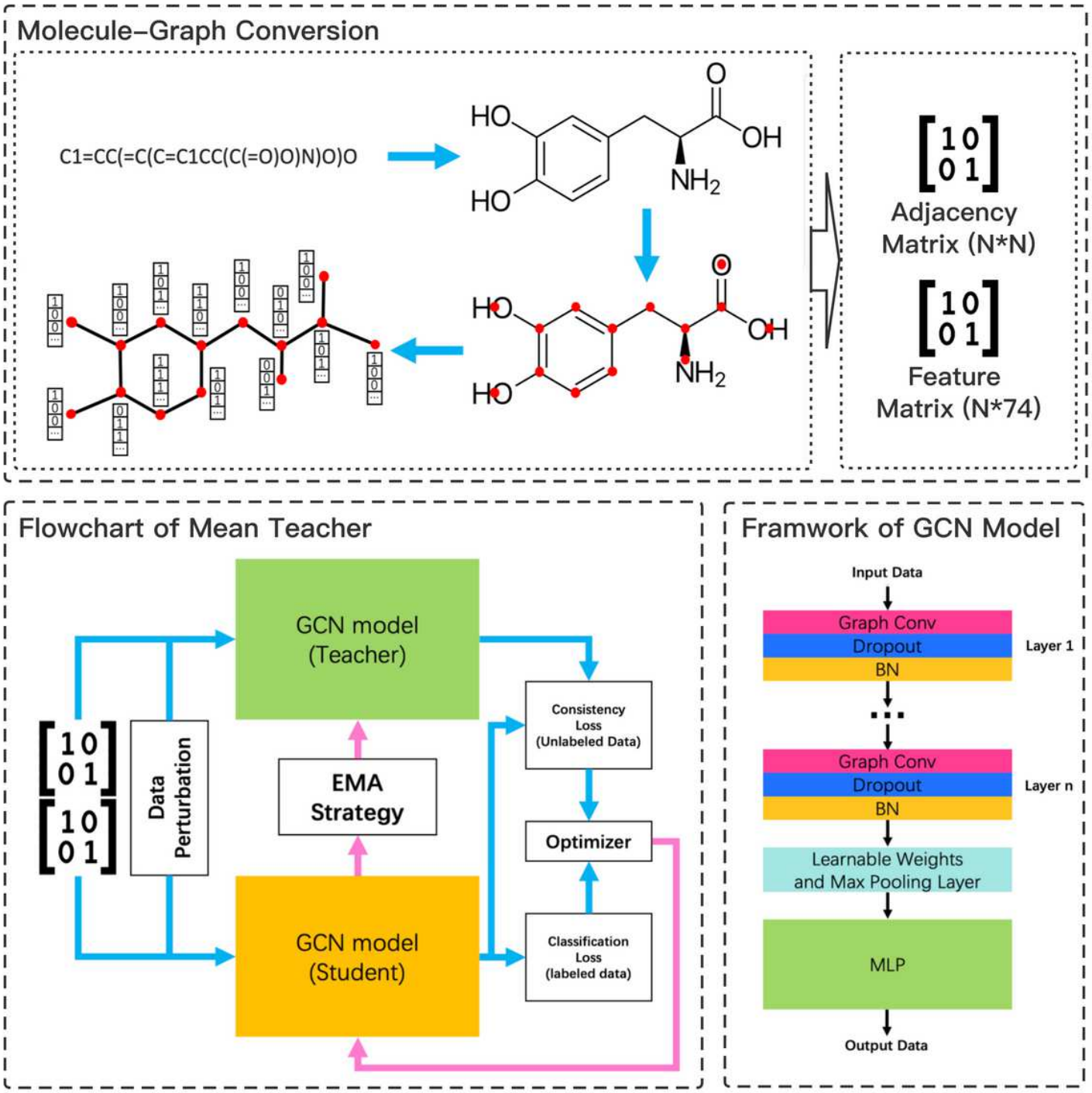
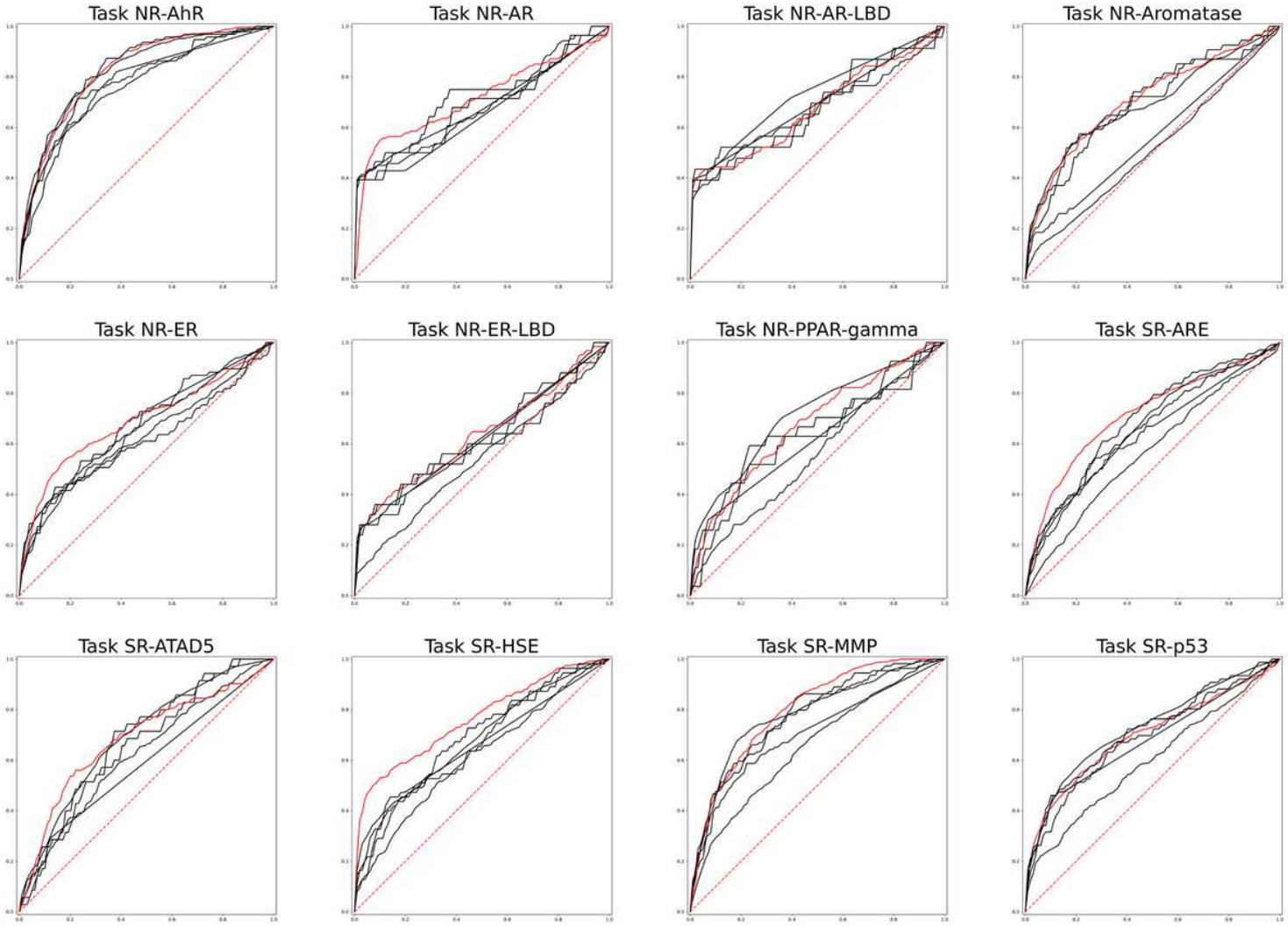



**Figure 1**

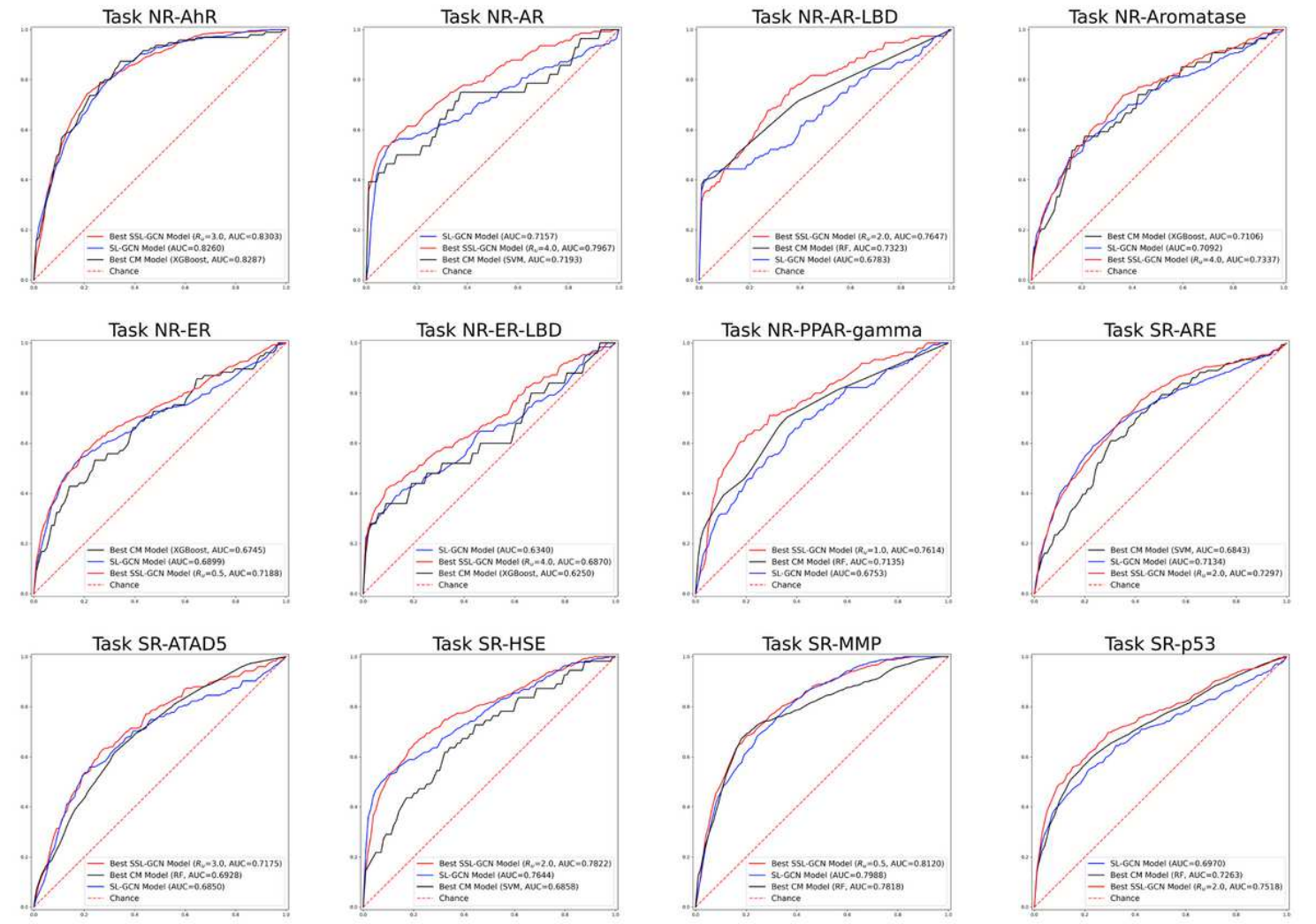
The SSL-GCN model for compound toxicity prediction. Molecular compounds are converted into graphs of nodes and connections. The GCN model architecture is composed of two stacked layers of graph convolutional layer, dropout, and batch normalization layer. All signals are summarized by the max

pooling layer and fed into the multilayer perceptron network to generate the final output. The teacher and student GCN models are updated using the MT algorithm.




## Figure 2

ROC curves of conventional ML models and SL-GCN models. The comparison of ROC curves between conventional ML models (black line) and SL-GCN models (red line) on 12 toxicity prediction tasks. Additional information of the ROC curves are provided in the supplementary




## Figure 3

ROC curves of best SSL-GCN, SL-GCN, and CM models. The comparison of ROC curves between the best conventional ML models (CM, black line), SL-GCN models (blue line), and SSL-GCN models with the best Ru (red line) on 12 toxicity prediction tasks. Additional information on the ROC curves can be found in the supplementary information.

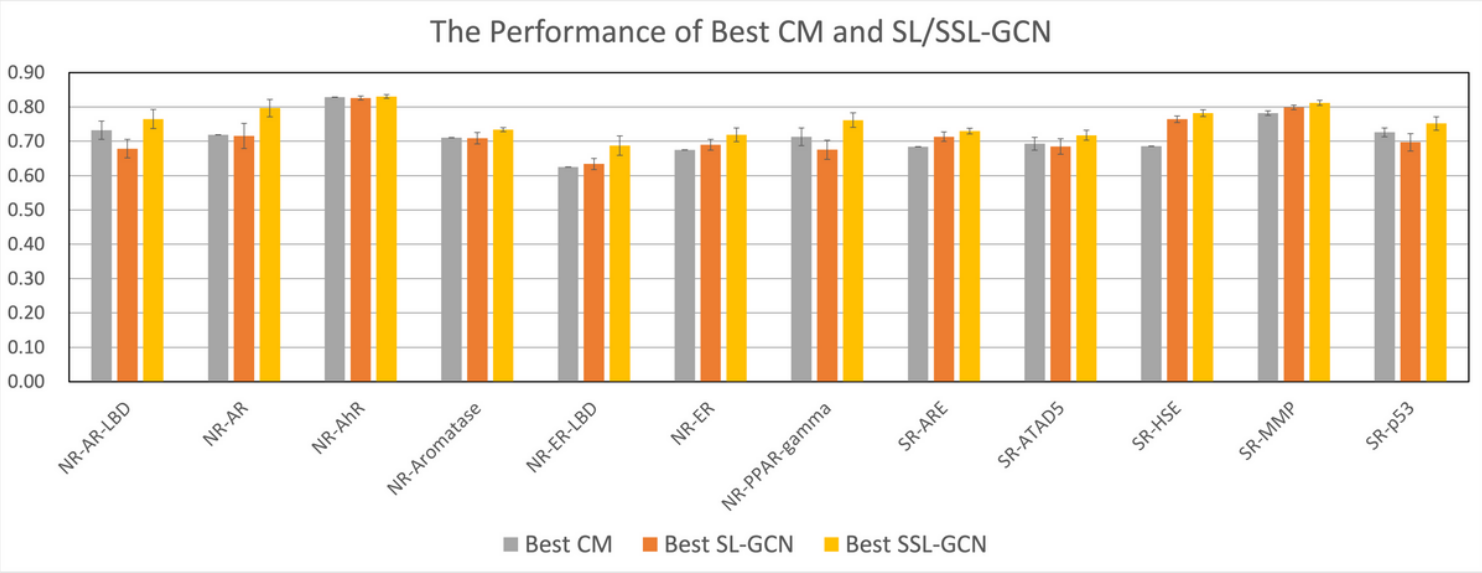



**Figure 4**

Comparison of AUC scores between SL-GCN, SSL-GCN and CM models Comparison of the best models from conventional methods (CM), SL-GCN, and the SSL-GCN on twelve toxicity prediction tasks. The mean and standard deviation are obtained from the 5-repeat experiments.

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupportingInformation.pdf