METHODOLOGY

MDDI-SCL: predicting multi-type drug-drug interactions via supervised contrastive learning

Shenggeng Lin[1], Weizhi Chen[1], Gengwang Chen[1], Songchi Zhou[1], Dong-Qing Wei[1,2,3*] and Yi Xiong[1,4*]

Abstract

The joint use of multiple drugs may cause unintended drug-drug interactions (DDIs) and result in adverse consequence to the patients. Accurate identification of DDI types can not only provide hints to avoid these accidental events, but also elaborate the underlying mechanisms by how DDIs occur. Several computational methods have been proposed for multi-type DDI prediction, but room remains for improvement in prediction performance. In this study, we propose a supervised contrastive learning based method, MDDI-SCL, implemented by three-level loss functions, to predict multi-type DDIs. MDDI-SCL is mainly composed of three modules: drug feature encoder and mean squared error loss module, drug latent feature fusion and supervised contrastive loss module, multi-type DDI prediction and classification loss module. The drug feature encoder and mean squared error loss module uses self-attention mechanism and autoencoder to learn drug-level latent features. The drug latent feature fusion and supervised contrastive loss module uses multi-scale feature fusion to learn drug pair-level latent features. The prediction and classification loss module predicts DDI types of each drug pair. We evaluate MDDI-SCL on three different tasks of two datasets. Experimental results demonstrate that MDDI-SCL achieves better or comparable performance as the state-of-the-art methods. Furthermore, the effectiveness of supervised contrastive learning is validated by ablation experiment, and the feasibility of MDDI-SCL is supported by case studies. The source codes are available at https://github.com/ShenggengLin/MDDI-SCL.

Keywords: Drug-drug interaction, Multi-type classification, Supervised contrastive learning, Multi-scale feature fusion, Self-attention mechanism

Introduction

The use of multiple drugs, often termed as polypharmacy, is a therapeutic approach to treat various complex diseases [1, 2]. However, polypharmacy can lead to drug-drug interactions (DDIs), in which the pharmacological effect of a drug is altered by another drugs [3–5]. It has been estimated that DDIs are associated with 30% of all the reported adverse drug events (ADEs) which may result in the majority of incidence and mortality, and even drug withdrawal from the market, incurring huge medical expense due to the stringent demands on drug development [6]. Therefore, it is necessary to reliably identify DDIs and understand their underlying mechanisms, which will be beneficial for drug development in pharmaceutical companies and can provide important information on polypharmacy prescription for clinicians and patients. In vitro experiments and clinical trials can be conducted to identify DDIs, but systematic combinatorial screening of DDI candidates from a large pool of drugs by experimental techniques remains challenging, time- and resource-consuming.

In the last decades, there are increasing availability of scientific literature, electronic medical records,

*Correspondence: dqwei@sjtu.edu.cn; xiongyi@sjtu.edu.cn

[1] State Key Laboratory of Microbial Metabolism, Shanghai-Islamabad-Belgrade Joint Innovation Center on Antibacterial Resistances, Joint International Research Laboratory of Metabolic & Developmental Sciences and School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China
Full list of author information is available at the end of the article

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 2 of 12

population-based reports of adverse events, drug labels, and other related sources [7]. Researchers attempted to extract DDIs from scientific literature and electronic medical records via natural language processing (NLP) techniques [8, 9], infer potential DDIs by similarity-based methods based on known DDIs [10], and predict DDIs by leveraging machine learning [11], network modelling [12, 13], and knowledge graphs [14, 15]. However, most of these computational methods (except the extraction of DDIs via NLP methods) only consider whether a DDI occurs or not given a pair of drugs.

To facilitate the understanding of the causal mechanisms of DDIs, recent studies have developed multi-type DDIs prediction methods to elaborate sufficient details beyond the chance of DDI occurrence [16]. The pioneering study by Ryu et al. constructed the gold standard DDI dataset from DrugBank [17], which covers 192,284 DDIs associated with 86 DDI types (changes in pharmacological effects and/or the risk of ADEs as a result of DDI) from 191,878 drug pairs [18]. Then, they formulated the multi-type DDI prediction as a multi-label classification task and proposed DeepDDI by using deep neural network (DNN) based on structural information of chemical compounds for a drug pair. This architecture became a baseline for several other state-of-the-art multi-type DDI prediction methods, which improved the multi-type DDI prediction by incorporating various types of biological information such as drug targets and enzymes to represent a drug pair in addition to the structural information of drugs based on autoencoder or the encoder module of transformer for learning the low-dimensional latent features and DNN algorithms for classification [19–21]. It should be noted that those methods represent the feature vector of a drug by the similarity profile, which is generated by the similarity (i.e., structural similarity) of a given drug against each one in the rest of drugs across the entire dataset. More recently, Deng et al. used few-shot learning based on the latent features from a pair of drug structures to improve the prediction performance on rare types of DDIs which have few samples [22]. Liu et al. proposed the method CSMDDI, which first generates the embedding representations of drugs and DDI types and then learns a mapping function to bridge the drugs attributes to their embeddings to predict multi-type DDIs [23]. Feng et al. proposed deep-MDDI, which consists of an encoder by deep relational graph convolutional networks constraining with similarity regularization to capture the topological features of DDI network and a tensor-like decoder for multi-label prediction of DDI types [24]. Yang et al. proposed a substructure-aware graph neural network, utilizing a message-passing neural network with a novel substructure attention mechanism and a substructure-substructure interaction module for DDI prediction [25].

With the increasing availability of large biomedical knowledge graphs (KGs), some studies attempt to incorporate KG with other data (i.e., drug molecular structures) for multi-type DDI predictions via graph neural networks (GNNs) [26, 27]. However, there are data redundancy and noise in the large KGs, in which only a small subgraph is relevant to a prediction target [28, 29]. Thus, the KG-based prediction methods for DDIs are still at the infant stage.

Although these published methods have achieved some success in multi-type DDI prediction, there still exist some limitations. First, datasets of DDI types are extremely unbalanced, and these methods have poor performance in predicting rare types with fewer samples. Second, most methods perform well in predicting unknown DDI types between known drugs, but they often fail to do it for new drugs. It will be useful to develop the new methods to resolve the problems and further improve the prediction performance.

Since the labelled data is limited and expensive to obtain, contrastive learning has recently become a popular and powerful strategy to get quality representations of samples in a self-supervised way. It aims at embedding augmented versions of the same sample close to each other while trying to push away embeddings from different samples [30]. Contrastive learning is not only used for self-supervised tasks, but also for supervised tasks. Khosla et al. extend the self-supervised batch contrastive approach to the fully-supervised setting, allowing models to effectively leverage label information [31]. For supervised contrastive learning, the samples belonging to the same class are pulled together in embedding space, while simultaneously pushing apart samples from different classes [31, 32].

Contrastive learning has been successfully applied in the field of bioinformatics [33–38]. In this study, we propose a new method named MDDI-SCL for multi-type DDI prediction, which is based on Supervised Contrastive Learning (SCL) and three-level loss functions. MDDI-SCL (Fig. 1) mainly includes three parts: drug feature encoder and mean squared error (MSE) loss module, drug latent feature fusion and supervised contrastive loss module, DDI type prediction and classification loss module. Specifically, we first input the drugs into the drug encoder to obtain the lower-dimensional latent features of each drug by MSE. Then, the latent features of two drugs are combined as input into the feature fusion module to obtain the latent features of the drug pairs. Supervised contrastive loss can make the features of the same

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 3 of 12

Fig. 1 The overview of the proposed MDDI-SCL method. **A** Drug feature encode and MSE loss module. **B** Drug latent feature fusion and supervised contrastive loss module. **C** Multi-type DDIs prediction and classification loss module. **D** Multi-head Attention (ATT) module. **E** Dense layer module
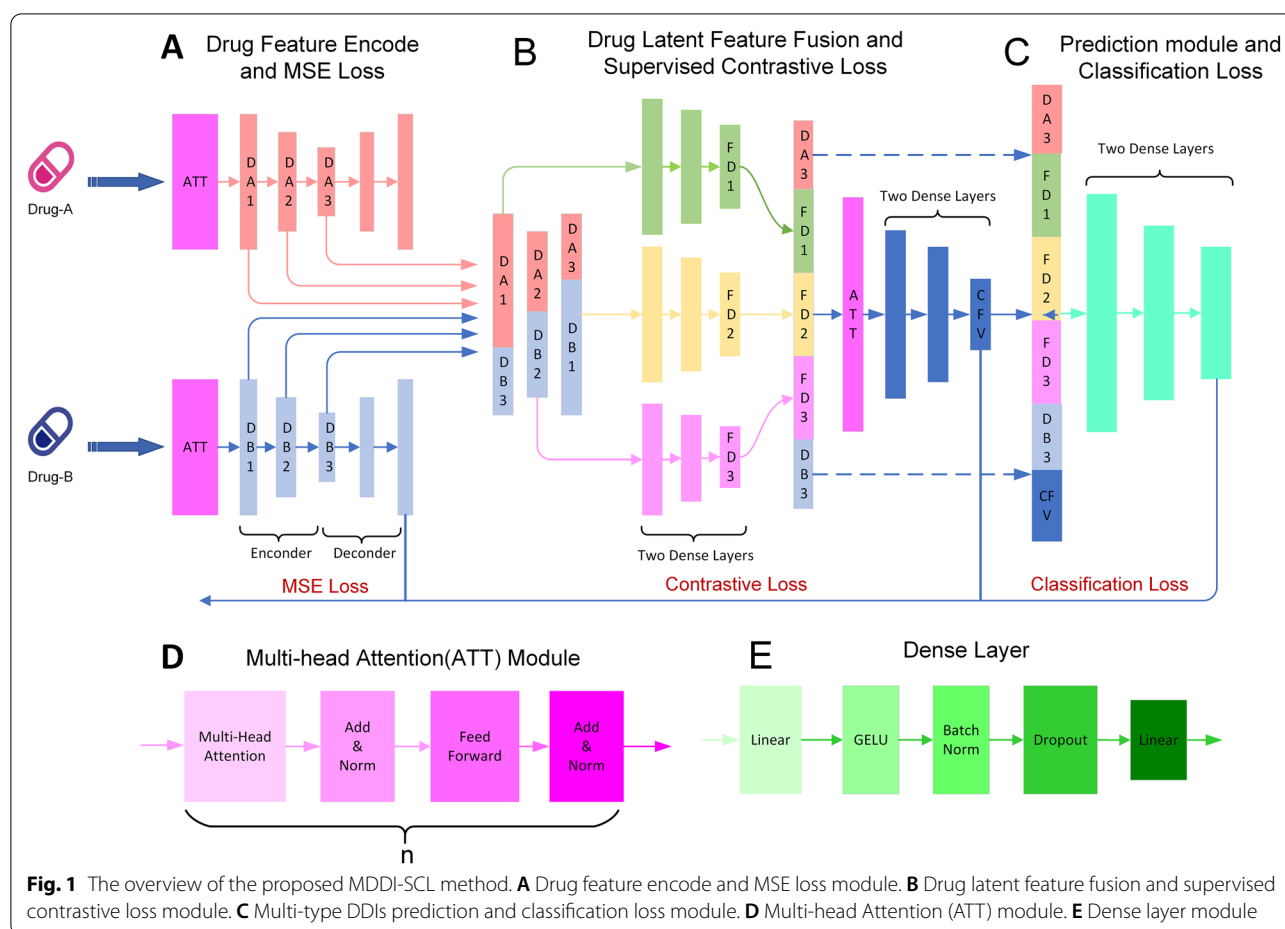
type of DDIs more similar, and the features of DDIs from different types more different. Therefore, we can obtain features that are more powerful to classification by using contrastive loss in the feature fusion module. Finally, we input the latent features of each drug pair into the multi-type DDI prediction module to predict DDI types, and update the model parameters by the classification loss.

Experimental results demonstrate that MDDI-SCL achieves better performance than several state-of-the-art methods on all three tasks of two different datasets. Additionally, we also proved the effectiveness of supervised contrastive learning for multi-type DDI prediction. More importantly, results of the case studies validated the feasibility of our method in practice.

Materials and methods
Datasets
In this study, we use two datasets with the number of samples at a different scale. The first dataset (Dataset1) is the benchmark dataset that Deng et al. collected [20]. Dataset1 contains 572 drugs with 74, 528 pairwise DDIs, which are associated with 65 DDI types. Each drug in Dataset1 has four types of features: chemical substructures, targets, pathways and enzymes, which are extracted from DrugBank [39]. The second dataset (Dataset2) is the dataset from the study of Lin et al. [21]. Dataset2 contains 1, 258 drugs with 323, 539 pairwise DDIs, which are associated with 100 DDI types. Each drug in Dataset2 has three types of features: substructures, targets and enzymes.

Drug feature representation
Each feature type of a drug corresponds to a set of descriptors, so one drug can be represented by a binary feature vector, and its value (1 or 0) indicates the presence or absence of the corresponding element.

These feature vectors have high dimensionality with values of most of dimensions being 0. Therefore, we represent the feature vector of a drug by the similarity

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 4 of 12

profile, which is generated by the similarity of drug A against each one (i.e., drug B) in the rest of drugs in the dataset [18]. Jaccard similarity is calculated by the following equation,

$$\mathbf{Jaccard(A, B)} = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|} = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A}| + |\mathbf{B}| - |\mathbf{A} \cap \mathbf{B}|} \quad (1)$$

where A and B are original bit vectors of two drugs; |A∩B| is the number of elements in the intersection of A and B; |A∪B| is the number of elements in the union of A and B.

Based on the Jaccard similarity, in Dataset1, each type feature of a drug is represented as a 572-dimensional vector. Therefore, each drug with four type of features is represented by a 4*572-dimentional vector. In the similar way, each drug is represented as a 3*1258-dimensional vector in Dataset2.

Drug feature encoder and mean squared error loss

The drug feature encoder module mainly includes multi-head self-attention layers and an autoencoder. The multi-head self-attention layers can focus on more important drug features [40, 41], and further the autoencoder performs feature dimensionality reduction [42, 43]. Consequently, lower-dimensional and better drug representations can be obtained through the drug feature encoder module. We use mean squared error loss to update the parameters of the feature encoder module.

Multi-head self-attention mechanism and autoencoder

The detailed description of the multi-head self-attention mechanism and autoencoder is provided in the Additional file 1 [41]. In the model, the hidden features obtained through the multi-head self-attention layers are denoted as DA1 and DB1 for a pair of drugs (i.e., drug A and drug B), as shown in Fig. 1A. The encoder of autoencoder has two linear layers. The output vectors of the first linear layer are denoted as DA2 and DB2, and the output vectors of the second linear layer are denoted as DA3 and DB3.

Mean squared error

Mean squared error is commonly used as regression loss function, which calculates average squared difference between the observed and predicted values. In our model, MSE is the sum of squared distances between the drug feature vector and the output vector of decoder divided by the feature dimensionality. The MSE is calculated by following formula,

$$MSE = \frac{\sum_{i=1}^{fea\_dim} (val_i - val_i^{\sim})^2}{fea\_dim} \quad (2)$$

where *fea_dim* is the feature dimensionality of the drug, val_i is the value of each dimension of the drug feature vector, $val_i^{\sim}$ is the value of each dimension of the output vector of the decoder.

Drug latent feature fusion and supervised contrastive loss

The drug latent feature fusion module mainly includes two sub-modules: multi-scale feature fusion and latent feature dimensionality reduction. The multi-scale feature fusion sub-module can simultaneously combine the low-level features and high-level features of a drug pair, and the feature dimensionality reduction sub-module can further fuse latent features and reduce the feature dimensionality. The supervised contrastive learning loss function is utilized to update the parameters of the drug latent feature fusion module.

Multi-scale feature fusion sub-module

A drug pair contains two drugs (i.e., drug A and drug B). Through the drug feature encoder module, three latent features of drug A are obtained: DA1, DA2, and DA3, as shown in Fig. 1A. Similarly, we can acquire three latent features of drug B: DB1, DB2, and DB3. DA1 and DB1 are low-level features, which usually contain more detailed information but also more noise [44, 45]. DA3 and DB3 are high-level features. Normally, high-level features have more semantic information and less noise but lose a lot of detailed information [45–48]. Thus, in order to better integrate the advantages of low-level features and high-level features, we concatenate DA1 and DB3, DA2 and DB2, DA3 and DB1 to represent a drug pair, respectively. Then, we input the concatenated features into the fully connected layer to obtain the fused drug pair features FD1, FD2, and FD3, as shown in Fig. 1B.

Latent feature dimension reduction sub-module

When the neural network becomes deep, residual connection can be used to avoid the problem of vanishing gradient [49]. In this sub-module, the output (DA3 and DB3) of encoder and the output (FD1, FD2 and FD3) of multi-scale feature fusion sub-module are concatenated as input into the latent feature dimensionality reduction sub-module, which mainly includes multi-head self-attention layers and linear layers. The number of neurons for each linear layer is half of the former layer. Multi-head self-attention has been introduced in detail in "Multi-head self-attention mechanism and autoencoder" section. The output vector of latent feature

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 5 of 12

dimensionality reduction sub-module is named CFV, as shown in Fig. 1B.

Supervised contrastive loss

Contrastive learning includes unsupervised contrastive learning and supervised contrastive learning. The latent features of samples obtained by unsupervised contrastive learning have the following property: the features of samples from the same source are more similar, whereas the features of samples from different sources are more different [50]. However, one significant disadvantage of unsupervised contrastive learning is that it does not consider the correlation of features between samples from different sources yet belonging to the same class. To overcome this drawback of unsupervised contrastive learning, supervised contrastive learning is proposed. The latent features of samples obtained by supervised contrastive learning have the following property: the features of samples belonging to same type are more similar, while the features of samples of different types are more different [31, 51].

Considering that the DDI type prediction task is a multi-class classification task, supervised contrastive learning is more competent for this task. Accordingly, our model employs supervised contrastive learning. The loss function of supervised comparative learning in our model can be calculated by the following formula,

$$l^{con} = \frac{1}{N_{batchsize}} \sum_{i=1}^{N_{batchsize}} l_i^{con} \tag{3}$$

Multi-type DDI prediction and classification loss

The module employs two fully connected layers to predict DDI types, and the number of neurons in the second fully connected layer is the number of DDI types. DDI type prediction is a multi-class classification task, and the sample size of each class is not balanced. Since focal loss can partially solve the problem of sample imbalance [21], we use focal loss [52] and cross-entropy loss as our classification loss functions. In detail, we choose the cross-entropy loss as our classification loss function in the first one third of training steps, and apply focal loss as our classification loss function in the last two thirds of steps. Therefore, the total loss function of the model is as follows:

$$Loss = l_{MSE}(x, \tilde{x}) + l_{con}(CFV, y) + l_{cla}(y, \tilde{y}) \tag{5}$$

, where x is the feature vector of the drug pair, $\tilde{x}$ is the output vector of the decoder, *CFV* is the output vector of latent feature dimensionality reduction sub-module, y is the class label of sample, and $\tilde{y}$ is the predicted value of sample. l_{MSE} is MSE loss function, l_{con} is supervised contrastive learning loss function and l_{cla} is classification loss function. l_{cla} is composed of the cross-entropy loss in the first one third of training steps and focal loss in the last two thirds of steps.

In order to prevent over-fitting, the label smoothing strategy is implemented [53]. For multi-classification problems, the class label vector is often converted into one-hot vector. However, the one-hot vector may weaken the generalization ability of the model and result in over-fitting. Label smoothing uses the smoothing parameter

$$l_i^{con} = \frac{-1}{N_{y_i} - 1} \sum_{j=1, j \neq i, y_j = y_i}^{N_{batchsize}} \log \frac{exp(sim(CFV_i, CFV_j)/\tau)}{\sum_{k=1, k \neq i}^{N_{batchsize}} exp(sim(CFV_i, CFV_k)/\tau)} \tag{4}$$

where $N_{batchsize}$ is the number of samples in each batch, y_i is the class label of sample i, and y_j is the class label of sample j. N_{y_i} is the number of samples of class y_i in the same batch. *sim* is a function that measures the similarity of two vectors, such as cosine similarity. CFV_i, CFV_j, CFV_k are the latent feature vector, which are the output vector of latent feature dimensionality reduction sub-module of sample i, j, and k, respectively. $\tau \in R^+$ is a scalar temperature parameter. According to the above formulas, in order to make the l_i^{con} loss smaller, the value of $sim(CFV_i, CFV_j)$ will be larger. So the hidden vectors CFV_i and CFV_j must be more similar. CFV_i and CFV_j are the latent vectors of the same type samples, so the latent features of the same type samples are more similar.

to add noise to the one-hot encoding, making the model less confident about its predictions. Therefore, it can partially solve the problem of over-fitting.

We utilize Gaussian error linear unit activation function and Radam optimizer [54]. The dropout layer and batch normalization layer are placed between the fully connected layers [55].

Results and discussion

Experimental settings of prediction tasks

This study evaluated the multi-type DDI prediction tasks based on three experimental settings: (i) prediction of unobserved interaction types between known drugs

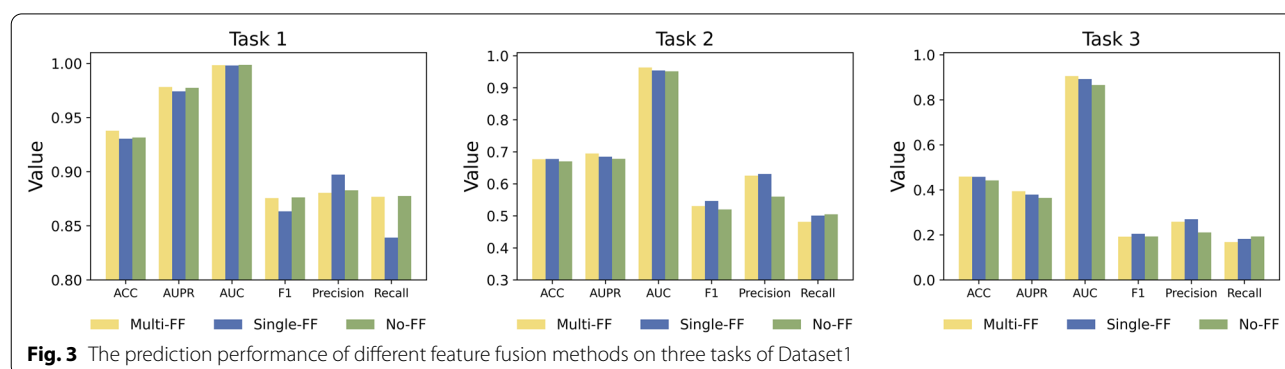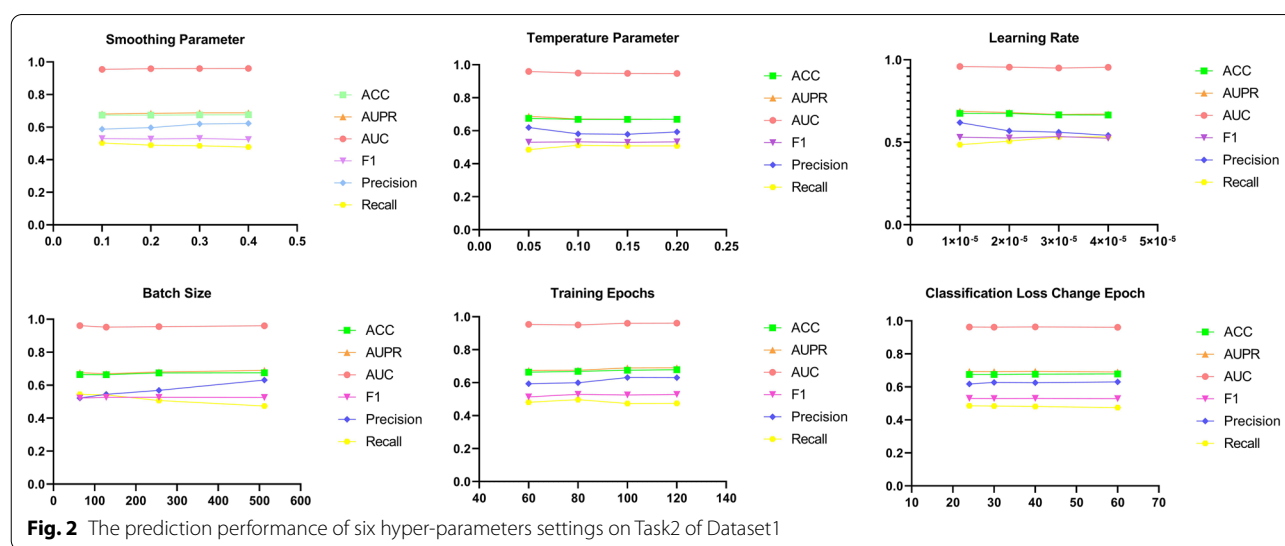Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 6 of 12

(Task1); (ii) prediction of interaction types between known drugs and new drugs (Task2) and (iii) prediction of interaction types between new drugs (Task3). New drugs in the corresponding task are missing in the training set, but exist in the test set.

For Task1, we apply five-fold cross-validation (5-CV) to DDI types and split all DDI types into five subsets. We train models based on DDI types in the training set, and then make predictions for DDI types in the test set. For Task2 and Task3, we apply 5-CV to drugs instead of DDI types. We randomly split drugs into five subsets, and used four of them as training drugs, leaving the remaining one as test drugs. For Task2, prediction models are constructed on the DDI types between two training drugs, and then make predictions for DDI types between training drugs and test drugs. For Task3, prediction models are built on the DDI types between two drugs in the training set to predict for DDI types between two drugs in the test set.

For model evaluation, accuracy (ACC), area under the precision-recall-curve (AUPR), area under the ROC curve (AUC), F1 score, precision and recall are adopted as evaluation metrics. On highly imbalanced data sets, AUPR and F1 score metrics are more objective for model evaluation. Consequently, in the following discussion, we will focus on these two metrics.

Hyper-parameters setting

The chosen of hyper-parameters influences the performance of model. First, we discussed the settings of six hyper-parameters on affecting the prediction performance on Task2 of Dataset1: smoothing parameter in the label smoothing strategy, temperature parameter in the contrastive learning, learning rate, batch size, training epochs and the epoch to change the cross-entropy loss to focal loss. Task1 is a relatively simple task, while Task3 is a relatively difficult task. Thus, to ensure the versatility of the hyper-parameters, we chose Task2 to tune

Fig. 2 The prediction performance of six hyper-parameters settings on Task2 of Dataset1

Fig. 3 The prediction performance of different feature fusion methods on three tasks of Dataset1

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 7 of 12

the hyper-parameters. For Task1 and Task3, we used the optimal parameters tuned on Task2. The performance metrics under different settings are shown in Fig. 2.

According to Fig. 2, the performance of the model does not change drastically as the hyper-parameters change. Almost all metric scores vary within the range of 0.01. This also illustrates the stability of our model. In the end, we chose 0.3 for smoothing parameter, 0.05 for temperature parameter, 2e-5 for learning rate, 512 for batch size, 120 for training epochs and the 40th epoch to change the cross-entropy loss to focal loss.

The prediction effect of multi-scale feature fusion
In the drug latent feature fusion module, we tried three types of feature fusion methods. The first method is the single-scale feature fusion, which concatenates DA1 and DB1, DA2 and DB2, DA3 and DB3 as three assemblies.

The second method is multi-scale feature fusion. Correspondingly, we concatenate DA1 and DB3, DA2 and DB2, DA3 and DB1 as three assemblies. The third method is to use only DA3 and DB3 without feature fusion. We compared these three feature fusion methods on three tasks of Dataset1, as shown in Fig. 3.
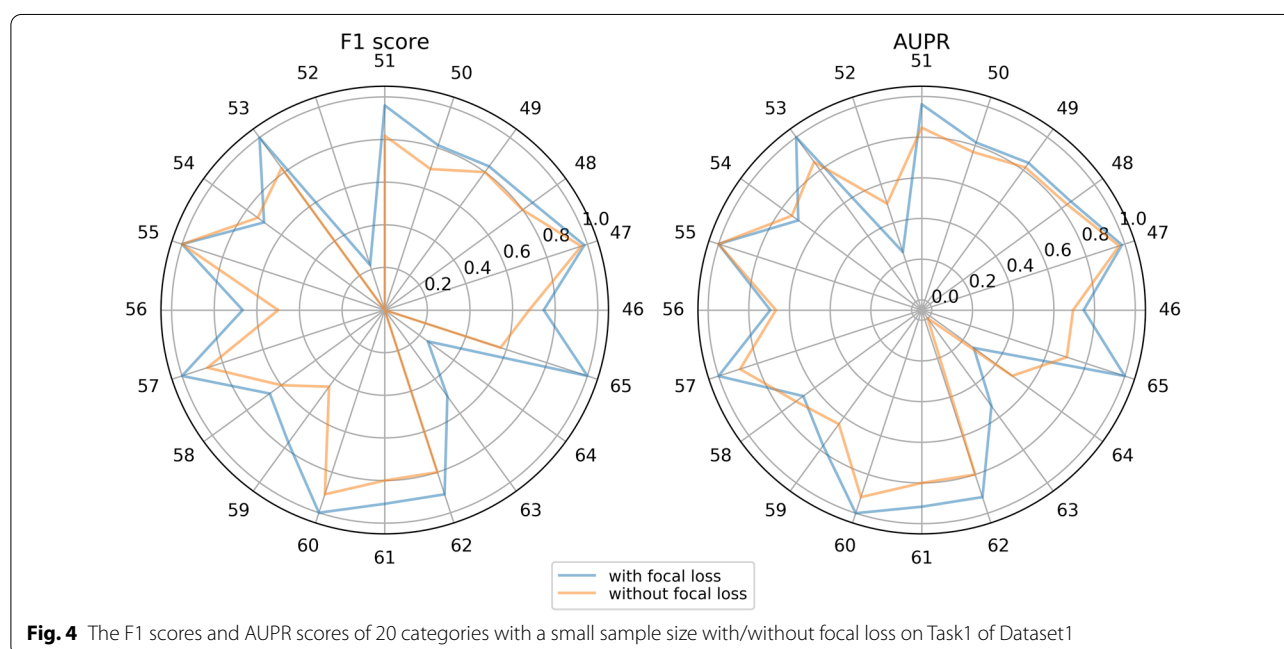
On three tasks, the AUPR and AUC of the multi-scale feature fusion method achieved the highest scores. In general, the performance of the multi-scale feature fusion method is slightly better than the other two methods. Therefore, multi-scale feature fusion is incorporated into the final model.

The prediction effect of supervised contrastive learning
In order to verify the effectiveness of supervised contrastive learning, we compared the performance of the model with and without supervised contrastive learning on

Table 1 The prediction effect of supervised contrastive learning on three tasks of Dataset1

	ACC	AUPR	AUC	F1	Precision	Recall
Task1						
With SCL	0.9378	0.9782	0.9983	0.8755	0.8804	0.8767
Without SCL	0.9308	0.9746	0.9982	0.8712	0.8762	0.8752
Task2						
With SCL	0.6767	0.6947	0.9634	0.5304	0.6254	0.4814
Without SCL	0.6667	0.6765	0.9513	0.5314	0.5685	0.5177
Task3						
With SCL	0.4589	0.3938	0.9053	0.1919	0.2585	0.1678
Without SCL	0.4553	0.3772	0.8740	0.2273	0.2571	0.2177

Fig. 4 The F1 scores and AUPR scores of 20 categories with a small sample size with/without focal loss on Task1 of Dataset1

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 8 of 12

three tasks of Dataset1, as shown in Table 1. The model with supervised contrastive learning achieved better performance in ACC, AUPR, and AUC on all three tasks. The AUPR of the model with supervised contrastive learning on Task2 is 0.6947 while the AUPR of the model without supervised contrastive learning on Task2 is 0.6765. The AUC of the model with supervised contrastive learning on Task3 is 0.0313 higher than that without supervised contrastive learning. In general, model with supervised contrastive learning achieves better prediction performance.

The prediction effect of focal loss

Focal loss can solve problems of imbalance in sample size of each category and difficulty of imbalanced classification. Focal loss improves the classification ability of the model by forcing the model to focus on categories with a small sample size. In order to examine whether focal loss improves the prediction for categories with small sample size, we selected 20 categories with the smallest sample size (from DDI type46 to DDI type65) on Task1 of Dataset1 for comparison, as shown in Fig. 4.

On categories with a small sample size, focal loss can boost the classification performance of the model. Among the 20 categories with a small sample size, the F1 score of the model with focal loss is higher than that of the model without focal loss on 19 categories. On DDI type 52, 63, and 64, the F1 score of the model without focal loss is 0, while the F1 score of the model with focal loss is 0.2222, 0.5, and 0.25, respectively. Among the 20 categories with a small sample size, the AUPR of the model with focal loss is higher than the AUPR of the model without focal loss on 16 categories. On DDI type 63, the AUPR of the model without focal loss is 0.0001, while the AUPR of the model with focal loss is 0.5334.

The prediction effect of label smoothing strategy

We verified the effectiveness of the label smoothing strategy on three tasks of Dataset1. The experimental results are shown in Table 2.

On all three tasks, the AUPR of the model using label smoothing is higher than that of the model which does not utilize label smoothing. The AUPR of the model using label smoothing on Task2 is 0.0242 higher than that without label smoothing. The AUPR of the model using label smoothing on Task3 is 0.0302 higher than that without label smoothing.

Comparison with state-of-the-art DDI type prediction and baseline methods

We compared MDDI-SCL with other four state-of-the-art DDI type prediction methods: DeepDDI [18], Lee et al.'s methods [19], DDIMDL [20] and MDF-SA-DDI [21], and also several baseline classification methods: fully connected DNN, random forest (RF), k-nearest neighbor (KNN) and logistic regression (LR). The performance comparison of all prediction models on Dataset1 and Dataset2 is shown in Table 3 and Table 4, respectively.

We evaluated the performance of all prediction methods for Task1. Experimental results show that MDDI-SCL and MDF-SA-DDI perform much better than other methods on Task1 on Dataset1. MDDI-SCL achieves the best AUPR 0.9782. On Dataset2, the performance of MDDI-SCL is better than other methods. The AUPR, F1 score and ACC of MDDI-SCL is 0.9862, 0.9321 and 0.9516, respectively. These evaluation scores of MDDI-SCL are higher than that of other methods.

We also compared the state-of-the-art methods on Task2 and Task3 of the two datasets. Experimental results show that our method MDDI-SCL achieves better or comparable performance than the state-of-the-art

Table 2 The prediction effect of label smoothing (LS) strategy on three tasks of Dataset1

	ACC	AUPR	AUC	F1	Precision	Recall
Task1						
With LS	0.9378	0.9782	0.9983	0.8755	0.8804	0.8767
Without LS	0.9377	0.9776	0.9981	0.8840	0.8718	0.9023
Task2						
With LS	0.6767	0.6947	0.9634	0.5304	0.6254	0.4814
Without LS	0.6659	0.6705	0.9470	0.5120	0.5275	0.5243
Task3						
With LS	0.4589	0.3938	0.9053	0.1919	0.2585	0.1678
Without LS	0.4449	0.3636	0.8723	0.1971	0.2022	0.2063

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 9 of 12

Table 3 Performance comparison with the state-of-the-art methods on three tasks of Dataset1

	ACC	AUPR	AUC	F1	Precision	Recall
Task1						
MDDI-SCL	0.9378	0.9782	0.9983	0.8755	0.8804	0.8767
MDF-SA-DDI	0.9301	0.9737	0.9989	0.8878	0.9085	0.8760
DDIMDL	0.8852	0.9208	0.9976	0.7585	0.8471	0.7182
Lee et al.'s methods	0.9094	0.9562	0.9961	0.8391	0.8509	0.8339
DeepDDI	0.8371	0.8899	0.9961	0.6848	0.7275	0.6611
DNN	0.8797	0.9134	0.9963	0.7223	0.8047	0.7027
RF	0.7775	0.8349	0.9956	0.5936	0.7893	0.5161
KNN	0.7214	0.7716	0.9813	0.4831	0.7174	0.4081
LR	0.7920	0.8400	0.9960	0.5948	0.7437	0.5236
Task2						
MDDI-SCL	0.6767	0.6947	0.9634	0.5304	0.6254	0.4814
MDF-SA-DDI	0.6633	0.6776	0.9497	0.5584	0.6547	0.5078
DDIMDL	0.6415	0.6558	0.9799	0.4460	0.5607	0.4319
Lee et al.'s methods	0.6405	0.6244	0.9247	0.5039	0.5388	0.4891
DeepDDI	0.5774	0.5594	0.9575	0.3416	0.3630	0.3890
DNN	0.6239	0.6361	0.9796	0.2997	0.4237	0.2840
Task3						
MDDI-SCL	0.4589	0.3938	0.9053	0.1919	0.2585	0.1678
MDF-SA-DDI	0.4338	0.3873	0.8630	0.2329	0.2715	0.2226
DDIMDL	0.4075	0.3635	0.9512	0.1590	0.2408	0.1452
Lee et al.'s methods	0.4097	0.3184	0.8302	0.2022	0.2216	0.2027
DeepDDI	0.3602	0.2781	0.9059	0.1373	0.1586	0.1450
DNN	0.4087	0.3776	0.9550	0.1152	0.1836	0.1093

methods on some evaluation metrics. On Dataset1, the AUPR of MDDI-SCL is 0.6947 and 0.3938 on Task2 and Task3, respectively. The AUC of MDDI-SCL is 0.6767 and 0.4589 on Task2 and Task3, respectively. These evaluation scores of MDDI-SCL are higher than that of other methods. The F1 score of MDDI-SCL is slightly worse than the state-of-the-art methods. It should be emphasized that we used the same hyper-parameters on different tasks and different datasets. We did not optimize the hyper-parameters of the model across all the datasets and tasks. The hyper-parameters of the deep learning model may affect the performance of the model, so the experimental results presented here may not be the optimal performance of our model.

In general, our model achieves better or similar performance on Task1 of both datasets compared to the state-of-the-art methods. Our model also achieves better or similar performance as the state-of-the-art methods on Task2 and Task3 of Dataset1. Our model performs slightly worse than the state-of-the-art models on Task2 and Task3 of Dataset2. This may be explained by the fact

that the hyper-parameters of our model are obtained on Dataset1. Inappropriate hyper-parameters may affect the performance of the model.

Case studies

The evaluation metrics have proved the effectiveness of our model. We conducted case studies to further validate the effectiveness of MDDI-SCL in practice.

We used all the DDI type samples on Dataset1 originally obtained from DrugBank [17] to train the prediction model, and then predicted the drug-drug pairs that do not exist on Dataset1. We focused on the five most frequent DDI types and checked up the top 20 predictions related to each type. We used the interactions checker tool provided by https://go.drugbank.com/drugs to verify these predictions.

Among 100 samples, 43 DDI type samples were confirmed, which are shown in Additional file 1: Table S1. For example, the interaction between Donepezil and Armodafinil is predicted to cause the DDI type #0, which

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 10 of 12

Table 4 Performance comparison with the state-of-the-art methods on three tasks of Dataset2

	ACC	AUPR	AUC	F1	Precision	Recall
Task1						
MDDI-SCL	0.9516	0.9862	0.9995	0.9321	0.9162	0.9500
MDF-SA-DDI	0.9291	0.9773	0.9996	0.9117	0.9381	0.8910
DDIMDL	0.9229	0.9637	0.9993	0.9105	0.9212	0.9039
Lee et al.'s methods	0.9370	0.9791	0.9991	0.9181	0.9226	0.9153
DeepDDI	0.7211	0.7724	0.9914	0.6854	0.6654	0.7183
DNN	0.7908	0.8539	0.9949	0.7649	0.7560	0.8046
RF	0.6956	0.7567	0.9892	0.5760	0.6694	0.5426
KNN	0.5797	0.5964	0.8998	0.3805	0.4758	0.3347
LR	0.5229	0.5288	0.9805	0.2373	0.3128	0.2185
Task2						
MDDI-SCL	0.6595	0.6794	0.9757	0.5578	0.5605	0.5712
MDF-SA-DDI	0.6664	0.6820	0.9862	0.5919	0.6526	0.5518
DDIMDL	0.6720	0.7086	0.9885	0.5817	0.6680	0.5295
Lee et al.'s methods	0.6917	0.7119	0.9687	0.5934	0.6144	0.5848
DeepDDI	0.5883	0.5851	0.9746	0.4709	0.5250	0.4361
DNN	0.6687	0.6838	0.9818	0.6164	0.7279	0.5479
Task3						
MDDI-SCL	0.4696	0.4261	0.9315	0.2838	0.3160	0.2773
MDF-SA-DDI	0.4794	0.4450	0.9686	0.2937	0.3667	0.2659
DDIMDL	0.4699	0.4386	0.9685	0.3032	0.3773	0.2729
Lee et al.'s methods	0.4867	0.4349	0.9093	0.3082	0.3355	0.3066
DeepDDI	0.3611	0.2820	0.9264	0.1868	0.2301	0.1711
DNN	0.4570	0.4129	0.9565	0.2997	0.4345	0.2508

means that metabolism of Donepezil can be decreased when combined with Armodafinil.

Under the same experimental setup, 43 of the 100 DDI samples predicted by MDDI-SCL were confirmed, whereas 35 of the 100 DDI samples predicted by MDF-SA-DDI were confirmed. This shows that MDDI-SCL is more effective than MDF-SA-DDI in practice. In Additional file 1: Table S2, we list the other 57 drug pairs among the 100 DDI samples. These drug pairs may not be reported in the literature, but these DDIs are likely to occur when taken together, which may be helpful for pharmaceutical research.

Conclusions

We proposed a multi-type DDI prediction model based on supervised contrastive learning and three-level loss functions, and proved the effectiveness and robustness of our model. In addition, we also proved the prediction effect of supervised contrastive learning, focal loss and label smoothing strategy. Experimental results demonstrate that our proposed model achieves better or comparable performance than that of the state-of-the-art models. The case studies were also performed to identify the new DDIs which are not included in the current datasets. Moreover, the effectiveness of our model is supported by case studies in practice.

Supplementary Information

> **Additional file 1: Table S1.** Forty-three DDI samples have been confirmed among the 100 DDI samples predicted by MDDI-SCL. **Table S2.** Fifty-seven DDI samples that may not be reported in the literature among the 100 DDI samples predicted by MDDI-SCL.

Availability of data and materials
The source codes are available at https://github.com/ShenggengLin/ MDDI-SCL. The datasets are available at https://github.com/ShenggengLin/ MDF-SA-DDI.

Declarations

Competing interests
The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author details
[1]State Key Laboratory of Microbial Metabolism, Shanghai-Islamabad-Belgrade Joint Innovation Center on Antibacterial Resistances, Joint International Research Laboratory of Metabolic & Developmental Sciences and School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China. [2]Zhongjing Research and Industrialization Institute of Chinese Medicine, Nanyang 473006, China. [3]Peng Cheng National Laboratory, Shenzhen 518055, China. [4]Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

References

1. Bansal M, Yang J, Karan C, Menden MP, Costello JC, Tang H, Xiao G, Li Y, Allen J, Zhong R et al (2014) A community computational challenge to predict the activity of pairs of compounds. Nat Biotechnol 32(12):1213–1222
2. Zitnik M, Agrawal M, Leskovec J (2018) Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 34(13):457–466
3. Vilar S, Harpaz R, Uriarte E, Santana L, Rabadan R, Friedman C (2012) Drug-drug interaction through molecular structure similarity analysis. J Am Med Inform Assn 19(6):1066–1074
4. Xiong G, Yang Z, Yi J, Wang N, Wang L, Zhu H, Wu C, Lu A, Chen X, Liu S et al (2022) DDInter: an online drug-drug interaction database towards improving clinical decision-making and patient safety. Nucleic Acids Res 50(D1):D1200–D1207
5. Su XR, Hu L, You ZH, Hu PW, Wang L, Zhao BW (2022) A deep learning method for repurposing antiviral drugs against new viruses via multi-view nonnegative matrix factorization and its application to SARS-CoV-2. Brief Bioinform. 23(1):bbab526
6. Tatonetti NP, Fernald GH, Altman RB (2012) A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports. J Am Med Inform Assoc 19(1):79–85
7. Tatonetti NP, Ye PP, Daneshjou R, Altman RB (2012) Data-driven prediction of drug effects and interactions. Sci Transl Med. https://doi.org/10.1126/ scitranslmed.3003377
8. Vilar S, Friedman C, Hripcsak G (2018) Detection of drug-drug interactions through data mining studies using clinical sources, scientific literature and social media. Brief Bioinform 19(5):863–877
9. Zhang Y, Zheng W, Lin H, Wang J, Yang Z, Dumontier M (2018) Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. Bioinformatics 34(5):828–835
10. Vilar S, Uriarte E, Santana L, Lorberbaum T, Hripcsak G, Friedman C, Tatonetti NP (2014) Similarity-based modeling in large-scale prediction of drug-drug interactions. Nat Protoc 9(9):2147–2163
11. Cheng F, Zhao Z (2014) Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. J Am Med Inform Assoc 21(e2):e278-286
12. Park K, Kim D, Ha S, Lee D (2015) Predicting pharmacodynamic drug-drug interactions through signaling propagation interference on protein-protein interaction networks. Plos ONE 10(10):e0140816
13. Zhang P, Wang F, Hu J, Sorrentino R (2015) Label propagation prediction of drug-drug interactions based on clinical side effects. Sci Rep 5:12339
14. Lin X, Quan Z, Wang ZJ, Ma TF, Zeng XX. KGNN: Knowledge Graph Neural Network for Drug-Drug Interaction Prediction. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence 2020:2739–2745.
15. Dai YF, Guo CH, Guo WZ, Eickhoff C (2021) Drug-drug interaction prediction with Wasserstein Adversarial Autoencoder-based knowledge graph embeddings. Brief Bioinform. 22(4):bbaa256
16. Zhang XD, Wang G, Meng XY, Wang S, Zhang Y, Rodriguez-Paton A, Wang JM, Wang X (2022) Molormer: a lightweight self-attention-based method focused on spatial structure of molecular graph for drug-drug interactions prediction. Brief Bioinform. 23(5):bbac296
17. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z et al (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res. 46(D1):D1074–D1082
18. Ryu JY, Kim HU, Lee SY (2018) Deep learning improves prediction of drug-drug and drug-food interactions. Proc Natl Acad Sci USA 115(18):E4304–E4311
19. Lee G, Park C, Ahn J (2019) Novel deep learning model for more accurate prediction of drug-drug interaction effects. BMC Bioinformatics. 20(1):415
20. Deng YF, Xu XR, Qiu Y, Xia JB, Zhang W, Liu SC (2020) A multimodal deep learning framework for predicting drug-drug interaction events. Bioinformatics 36(15):4316–4322
21. Lin SG, Wang YJ, Zhang LF, Chu YY, Liu YT, Fang YT, Jiang MM, Wang QK, Zhao BW, Xiong Y et al (2022) MDF-SA-DDI: predicting drug-drug interaction events based on multi-source drug fusion, multi-source feature fusion and transformer self-attention mechanism. Brief Bioinform. 23(1):bbab421
22. Deng Y, Qiu Y, Xu X, Liu S, Zhang Z, Zhu S, Zhang W (2022) META-DDIE: predicting drug-drug interaction events with few-shot learning. Brief Bioinform. 23(1):bbab421
23. Liu Z, Wang XN, Yu H, Shi JY, Dong WM (2022) Predict multi-type drug-drug interactions in cold start scenario. BMC Bioinformatics 23(1):75
24. Feng YH, Zhang SW, Zhang QQ, Zhang CH, Shi JY (2022) deepMDDI: a deep graph convolutional network framework for multi-label prediction of drug-drug interactions. Anal Biochem 646:114631
25. Yang ZD, Zhong WH, Lv QJ, Chen CYC (2022) Learning size-adaptive molecular substructures for explainable drug-drug interaction prediction by substructure-aware graph neural network. Chem Sci 13(29):8693–8703
26. Chen YJ, Ma TF, Yang XX, Wang JM, Song BS, Zeng XX (2021) MUFFIN: multi-scale feature fusion for drug-drug interaction prediction. Bioinformatics 37(17):2651–2658
27. Yu Y, Huang KX, Zhang C, Glass LM, Sun JM, Xiao C (2021) SumGNN: multi-typed drug interaction prediction via efficient knowledge graph summarization. Bioinformatics 37(18):2988–2995
28. Su XR, Hu L, You ZH, Hu PW, Zhao BW (2022) Attention-based knowledge graph representation learning for predicting drug-drug interactions. Brief Bioinform 23(3):bbac140
29. Hu L, Zhang J, Pan XY, Yan H, You ZH (2021) HiSCF: leveraging higher-order structures for clustering analysis in biological networks. Bioinformatics 37(4):542–550
30. Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F (2021) A survey on contrastive self-supervised learning. Technologies. https://doi.org/10. 3390/technologies9010002
31. Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, Maschinot A, Liu C, Krishnan D (2021) Supervised contrastive learning. arXiv. https://doi.org/ 10.48550/arXiv.2004.11362

Lin *et al. Journal of Cheminformatics*   (2022) 14:81

Page 12 of 12

32. Lopez-Martin M, Sanchez-Esguevillas A, Arribas JI, Carro B (2022) Supervised contrastive learning over prototype-label embeddings for network intrusion detection. Inform Fusion 79:200–228

33. Zheng L, Liu Z, Yang Y, Shen HB (2022) Accurate inference of gene regulatory interactions from spatial gene expression with deep contrastive learning. Bioinformatics 38(3):746–753

34. Liu X, Song C, Huang F, Fu H, Xiao W, Zhang W (2022) GraphCDR: a graph neural network method with contrastive learning for cancer drug response prediction. Brief Bioinform. 23(1):bbab457

35. Li Y, Qiao G, Gao X, Wang G (2022) Supervised graph co-contrastive learning for drug-target interaction prediction. Bioinformatics 38(10):2847–2854

36. Hu H, Bindu JP, Laskin J (2021) Self-supervised clustering of mass spectrometry imaging data using contrastive learning. Chem Sci 13(1):90–98

37. Wang YH, Min YS, Chen X, Wu J. Multi-view Graph Contrastive Representation Learning for Drug-Drug Interaction Prediction. Proceedings of the World Wide Web Conference 2021 (Www 2021) 2021:2921–2933.

38. Ciortan M, Defrance M (2021) Contrastive self-supervised clustering of scRNA-seq data. BMC Bioinformatics 22(1):280

39. Law V, Knox C, Djoumbou Y, Jewison T, Guo AC, Liu Y, Maciejewski A, Arndt D, Wilson M, Neveu V et al (2014) DrugBank 4.0: shedding new light on drug metabolism. Nucleic Acids Res. 42(Database issue):1091–1097

40. Chu YY, Zhang Y, Wang QK, Zhang LF, Wang XH, Wang YJ, Salahub DR, Xu Q, Wang JM, Jiang X et al (2022) A transformer-based model to predict peptide-HLA class I binding and optimize mutated peptides for vaccine design. Nat Mach Intell. 4(3):300

41. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neur In. https://doi.org/10.48550/arXiv.1706.03762

42. Dai QY, Chu YY, Li ZQ, Zhao YS, Mao XY, Wang YJ, Xiong Y, Wei DQ (2021) MDA-CF: predicting MiRNA-disease associations based on a cascade forest model by fusing multi-source information. Comput Biol Med. 136:104706

43. Rao JH, Zhou X, Lu YT, Zhao HY, Yang YD (2021) Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. iScience 24(5):102393

44. Liu S, Qi L, Qin HF, Shi JP, Jia JY. Path Aggregation Network for Instance Segmentation. 2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr) 2018:8759-8768

45. Singh B, Davis LS. An Analysis of Scale Invariance in Object Detection - SNIP. 2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr) 2018:3578-3587

46. Li YH, Chen YT, Wang NY, Zhang ZX. Scale-Aware Trident Networks for Object Detection. Ieee I Conf Comp Vis 2019:6053–6062.

47. Song T, Zhang XD, Ding M, Rodriguez-Paton A, Wang SD, Wang G (2022) DeepFusion: a deep learning based multi-scale feature fusion method for predicting drug-target interactions. Methods 204:269–277

48. Tang Q, Nie FL, Kang JJ, Chen W (2021) mRNALocater: enhance the prediction accuracy of eukaryotic mRNA subcellular localization by using model fusion strategy. Mol Ther 29(8):2617–2623

49. He KM, Zhang XY, Ren SQ, Sun J. Deep Residual Learning for Image Recognition. Proc Cvpr Ieee 2016:770–778.

50. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. Pr Mach Learn Res. 119:1597

51. Guo DE, Xia Y, Luo XB, Feng JF. 2021. Remote Sensing Image Scene Classification Based on Supervised Contrastive Learning. Acta Photonica Sinic. 50(7).

52. Lin TY, Goyal P, Girshick R, He KM, Dollar P (2020) Focal loss for dense object detection. Ieee T Pattern Anal 42(2):318–327

53. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. Proc Cvpr Ieee 2016:2818–2826.

54. Zheng W, Zhang YX, Gong XH, Zhanghuali, Yu BY. DenseNet model with RAdam optimization algorithm for cancer image classification. 2021 Ieee International Conference on Consumer Electronics and Computer Engineering (Iccece) 2021:771-775.

55. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv 37:448

Publisher's Note