Briefings in Bioinformatics, 2022, **23**, 1–12
https://doi.org/10.1093/bib/bbab545
Problem Solving Protocol

AMDE: a novel attention-mechanism-based multidimensional feature encoder for drug–drug interaction prediction

Shanchen Pang, Ying Zhang, Tao Song*, Xudong Zhang, Xun Wang and Alfonso Rodriguez-Patón

*Corresponding author. Tao Song, College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China. Tel: 15053258769; E-mail: t.song@upm.es

Abstract

The properties of the drug may be altered by the combination, which may cause unexpected drug–drug interactions (DDIs). Prediction of DDIs provides combination strategies of drugs for systematic and effective treatment. In most of deep learning-based methods for predicting DDI, encoded information about the drugs is insufficient in some extent, which limits the performances of DDIs prediction. In this work, we propose a novel attention-mechanism-based multidimensional feature encoder for DDIs prediction, namely attention-based multidimensional feature encoder (AMDE). Specifically, in AMDE, we encode drug features from multiple dimensions, including information from both Simplified Molecular-Input Line-Entry System sequence and atomic graph of the drug. Data experiments are conducted on DDI data set selected from Drugbank, involving a total of 34 282 DDI relationships with 17 141 positive DDI samples and 17 141 negative samples. Experimental results show that our AMDE performs better than some state-of-the-art baseline methods, including Random Forest, One-Dimension Convolutional Neural Networks, DeepDrug, Long Short-Term Memory, Seq2seq, Deepconv, DeepDDI, Graph Attention Networks and Knowledge Graph Neural Networks. In practice, we select a set of 150 drugs with 3723 DDIs, which are never appeared in training, validation and test sets. AMDE performs well in DDIs prediction task, with AUROC and AUPRC 0.981 and 0.975. As well, we use Torasemide (DB00214) as an example and predict the most likely drug to interact with it. The top 15 scores all have been reported with clear interactions in literatures.

Keywords: DDI prediction, multidimensional feature, encoder, deep learning

Introduction

It is known that most human diseases are caused by complex biological processes, which cannot be completely cured by any single drug. It needs to take multiple drugs at the same time for combination therapy. This treatment increases the possibility of drug–drug interactions (DDIs) and even adverse drug reactions [1–3]. Serious drug interactions may make the drug lose its therapeutic effect and may also lead to drug withdrawal [2]. Whether from the perspective of therapeutic benefit or economic benefit, it is very important to identify potential DDIs as soon as possible. However, the task faces many challenges. Although the clinical experiment is reliable, it has high cost, long cycle and low economic benefit [3–7].

With the development of artificial intelligence, machine learning methods can overcome the limitations of clinical experiment [8], so as to help scientists identify DDIs quickly and effectively [4].

Related works

Plenty of computational methods have been developed to identify DDIs. And the task of Identifying DDIs can be modeled as a binary classification task in [9–16]. The existing work focuses on two subtasks: encoding drug features and predicting interactions. Accurate prediction of DDIs depends on effective feature coding technology strongly [10–12]. Different feature coding technologies

Shanchen Pang graduated degree from the Tongji University of Computer Software and Theory, Shanghai, China, in 2008. He is now a Full Professor in the China University of Petroleum, Qingdao, China. His current research interests include theory and application of Petri Net, service computing and trusted computing.
Ying Zhang is a graduate student in the Department of Computer Science and Engineering at the China University of Petroleum, with a research interest in drug repositioning.
Tao Song received his PhD degree from the Huazhong University of Science and Technology, Wuhan, China, in 2013. He is currently a Full Professor in the China University of Petroleum, Qingdao, China, and a Juan de la Cierva supported researcher in the Polytechnic University of Madrid, Madrid, Spain. His current research interests include DNA computing, membrane computing and bioinformatics.
Xudong Zhang is a graduate student in the Department of Computer Science and Engineering at the China University of Petroleum, with a research interest in drug repositioning.
Xun Wang received her PhD degree from Tsukuba University, Tsukuba, Japan, in 2014. She is currently an Associated Professor in the China University of Petroleum, Qingdao, China, and a grand supported post-doc researcher in the China High Performance Computer Research Center, Institute of Computer Technology, Chinese Academy of Science, Beijing, China. Her current research interests include high-performance computing for bioinformatics.
Alfonso Rodriguez-Patón received the BS degree in electronic and computational from the University of Santiago de Compostela, Santiago de Compostela, Spain, in 1992, and the PhD degree in computer science from the Polytechnic University of Madrid, Madrid, Spain, in 1999. He is currently a Full Professor with the Faculty of Informatics, Polytechnic University of Madrid. His current research interests include interplay of computer science, biology and engineering.

Received: August 30, 2021. **Revised:** October 31, 2021. **Accepted:** November 26, 2021

may have different extent of deviations. In general, these technologies can be divided into three categories: chemical fingerprint-based, molecular graph-based and knowledge graph-based.

Traditional DDIs prediction models mostly use chemical fingerprint as input [9–14]. Chemical fingerprint is sequence data, which can be easily input into machine learning models. Chemical fingerprint can describe specific property of drug, such as substructure, related targets and side effects. One fingerprint cannot show all properties, so the methods based on chemical fingerprint often integrate multiple chemical fingerprints to predict DDIs [17–20]. These methods focus on extracting the features of sequence data. Natural language processing (NLP) methods can be used to process these fingerprint data, for example word2vec [17] and seq2seq [18].

Drugs are known as chemical molecules with spatial structure. No matter how many fingerprints are integrated, the sequence cannot show the spatial structure of drugs. Some important fingerprints are applicable to a small number of drugs, which limits the size of the data set. This kind of method has limited ability to encode the spatial structure of drugs and thus lack of scalability. Some other widely used methods to encode drug features are relying on molecular graph structure [19–23]. The input of the model is graph or data that can be converted into graph. In this way, adjacency matrix and feature matrix are used to represent the graph of drugs. The basic idea is to extract atomic information as drug features where atomic information is updated iteratively through Graph Neural Network (GNN). These methods can effectively encode the atomic information of drugs spatially.

In some GNNs, attention mechanism is added to enhance interpretability [20]. The physicochemical properties of drugs are generally manifested in specific substructures that are smaller than original drug. Encoding atomic features may destroy the chemical information represented by the substructure and also lose information about the chemical bonds connecting the atoms.

It is also popular to incorporate knowledge from multiple domains related to the drug-like protein targets and genes instead of solely focusing on domain-specific knowledge [24, 25]. It models the DDI prediction as a link prediction task. Firstly, a heterogeneous graph is constructed based on the relationships of drugs, proteins and genes. And then, the association of drugs in the heterogeneous graph is encoded as drug features. These models focus on the neighborhood relationships of the drug but ignore the drug's own structure. Although information in related domains is very useful for DDIs prediction, it is also very expensive to obtain [24, 26–29].

The physicochemical properties of drugs are complex in which single encoding of drug features from one-dimensional (1D) sequence or two-dimensional (2D) graphical structure cannot adequately represent the drug. Inadequate feature encoding can further eliminate the effectiveness of the DDIs prediction task.

In this work, we propose an attention-based multidimensional feature encoder (AMDE) to predict DDIs, considering the urgency of the DDIs prediction problems and the limitations of existing models. In a nutshell, we use Simplified Molecular-Input Line-Entry System (SMILES) string as input, which is a line notation that uses a predefined set of rules to describe the structure of compounds, which is sequential. Our model consists of three main modules. The first module is a 2D graph feature encoder, which transforms SMILES into atomic graph and uses Message Passing Attention Network (MPAN) [30] to extract both high-order structures and semantic relations of the graph. The representation of atomic graph is natural and rational because drugs are principally graph-structured with atoms as nodes and bonds as edges. The second module is a 1D sequence feature encoder, which splits SMILES into smaller length substructures and encodes the sequence features of drugs by Transformer [31]. This module can encode sequential relationships between substructures. The last module is a multidimensional feature decoder for predicting DDIs after hybridizing 2D and 1D features of drugs. Compared with previous studies, our contributions are summarized as follows:

(i) The attention-based multidimensional feature encoder is able to process the SMILES string of drugs from multiple dimensions. It encodes features that can represent the information of drugs more precisely.
(ii) The multidimensional feature decoder further compresses the drug feature vector and is able to strongly associate features with prediction results.
(iii) It is provided a new method of feature fusion: integrating drug features from multiple dimensions can enhance the effectiveness of downstream prediction tasks.

Model architecture

We formulate the DDIs prediction as a binary classification task to determine whether pairs of drugs would interact with each other. In our method, drugs are represented by a SMILES sequence, which consists of a sequence of chemical atoms and chemical bonds. The drug set is denoted as $D = \{d_1. \cdots, .d_n\}$; DDI predicted task can be modeled as a mapping function

$$\Delta : d_x \times d_y \xrightarrow{\Delta} \hat{P} \in \{0, 1\} . \qquad (1)$$

AMDE consists of two channels to extract both 2D graph features and 1D sequence features from the drug SMILES string simultaneously. The structure of the model is shown in Figure 1.

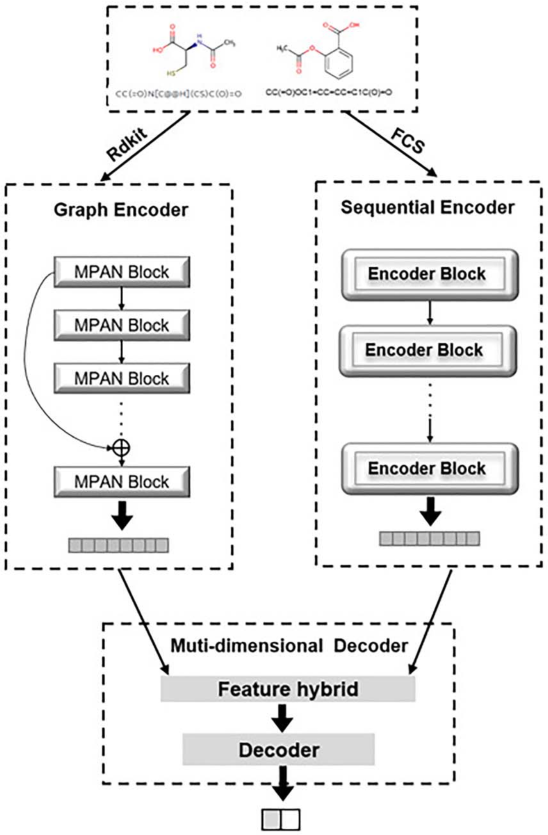

Figure 1. Structure of AMDE.

In Figure 1, the attention-based multidimensional feature encoder (AMDE) input is drug SMILES sequence. AMDE is divided into two channels to process the SMILES strings separately. The RDKit converts SMILES into atomic graph as input for the graph encoder. The graph encoder generates a 2D atomic graph feature vector of the drug through an MPAN. We use Frequent Consecutive Subsequence (FCS) Algorithm that encodes the SMILES directly to generate sequence data as input for the sequence encoder. The sequence encoder uses the Transformer to encode the sequence data directly to generate a feature vector that represents the 1D sequence of the drug. The 2D atomic graph feature vector and the 1D sequence feature vector are fed into multidimensional decoder. A higher dimensional vector is generated by feature hybrid part; then, a decoder is implemented on this vector to get a final token indicating whether DDI will occur.

Feature extraction

As shown in Figure 1, we use 2018.09.1.0-RDKit [32] to convert SMILES into undirected graph $G = \{V, E\}$. By symbol V, we denote the set of nodes and E is the set of edges. We consider the atoms as nodes and the bonds connecting the atoms as edges. We use the adjacency matrix $A \in R^{n\times n}$, the atomic feature matrix $N \in R^{n\times l}$ and the bond feature matrix$B \in R^{m\times k}$ to represent the graph structure of a drug ($n$denotes the number of atoms, $m$denotes the number of edges and l and k are the dimension of the feature). This representation can better present the internal structure information of the drug [33].

We also apply a data-driven sequential pattern mining algorithm called FCS Algorithm [34]. This algorithm is able to progressively decompose the ID sequence SMILES string of drugs into smaller subsequences and individual atom symbols. When traditional fingerprints based on substructure are used to represent drugs, the length of drug fingerprint (representing the number of substructures) is commonly over 100. Some of these substructures are still a subset of other substructures. Therefore, it is difficult to know which substructure leads to the outcome. FCS breaks the drug SMILES strings down into medium-sized substructures that are easier to give clear indication [34]. For a drugd_i, FCS results in an explicit substructure sequence:

$$\mathrm{FCS}\left(d_i\right) = S_i = \left\{s_1, s_2 \ldots s_p | s_k \in W\right\}, \tag{2}$$

where W is the FCS vocabulary; $S_i$denotes the FSC-encoded subsequences of d_i.

Graph feature encoder: MPAN

Message Passing Neural Network (MPNN) [35] is a kind of generalized GNN. MPNN is very suitable for extracting features of graph structured data. In recent years, MPNN has been used to solve molecular property prediction problems [30, 36–38]. We use MPAN, which is with the attention block added in MPNN, as a feature encoder for atomic graph in the DDIs prediction task [30].

The input of this module is 2D atomic graph $G = \{V, E\}$. Specifically, $V$is the set of nodes, containing the various atoms in the molecule $V = \{C, H, O \ldots\ldots\}$. $E$is the set of edges, which contains a total of four types. $E = \{e_{v,w} \in \mathrm{type} | v, w \in V\}$, type = {single, double, triple, aromatic bond.}.

Graph feature encoder is performed through the following two phases:

Phase1: Message passing. Nodes pass their own information in form of message vectors to other neighbor nodes along the edges of the graph, while nodes update the hidden features of itself by aggregating the message vectors passed from its neighbors. After K times message passing, each node receives message vectors from its Kth neighbors and the hidden features of each node are updated K times.

Phase2: Readout. After the hidden features of all nodes are updated, we use a readout function to aggregate the features of all nodes into the representation of the whole graph.

Message passing

As shown in Figure 2, each node initializes a fixed-size feature $h_v^{(0)} \in R^r$, which contains the chemical information of the atom itself [39] (such as atom type, valency, number of implicit H, number of electrons, type of hybridization and number of aromatic rings). Inputting

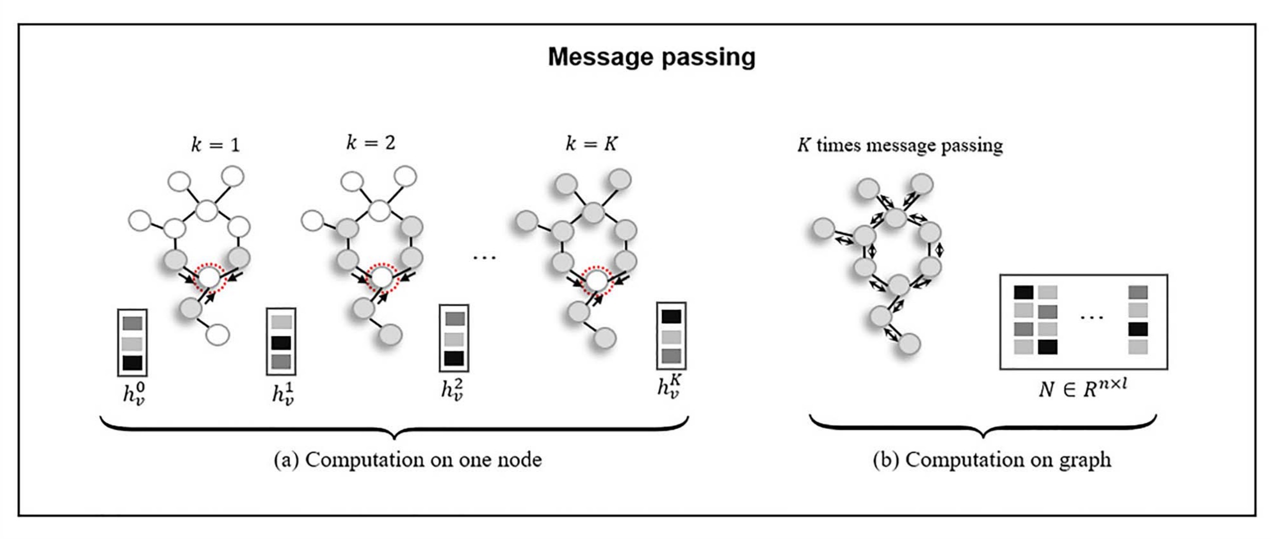

Figure 2. Message passing progress. (**A**) Computation on one node, the one marked by the red circle as an example has an initial feature of h_v^0. (**B**) Computation on graph. Message passing occurs at each atom, and each node is processed independently.

$h_v^{(0)}$ into a Convolutional Neural Network (CNN) [40] generates a node message vector

$$\text{message}_v^{(0)} = \text{CNN}\left(h_v^{(0)}\right). \tag{3}$$

The hidden features of nodes are updated iteratively along the edges of the graph by passing information between neighbor nodes. Thus, we define the aggregated message vector $m_v^{(k)}$ calculation step operated locally as

$$m_v^{(k)} = \text{Aggregation}\left(\text{message}_w^{(k)}, e_{v,w} | w \in N(v), e_{v,w} \in E\right), \tag{4}$$

where Aggregation denotes a message aggregating function, $N(v)$ denotes a self-included neighborhood of the node v and $e_{v,w}$ denotes the edge between v node and node w.

Notably, the message vector generated by the sender node is passed to the neighbor nodes by a specific type of edge. The receiving node aggregates messages from its neighbors, including information about the neighbor nodes and edges between the nodes. Then, each node updates the hidden features using its current hidden features $h_v^{(k)}$and the message from its neighbors $m_v^{(k)}$. This is completed according to following formulas:

$$h_v^{(k+1)} = \text{Update}\left(h_v^{(k)}, m_v^{(k)}\right). \tag{5}$$

When node updates its hidden features, we should focus on those neighbor nodes that provide critical information. Consequently, an attention block is added to this module. We calculate attention scores of neighbor nodes as weight coefficients in aggregating the neighboring message vectors. We define more precisely below

$$m_v^{(k)} = A_t\left(\text{message}_w^{(k)}, h_w^{(k)} e_{vw} | w \in N(v)\right) \tag{6}$$

$$A_k\left(\text{message}_w^{(k)}, h_w^{(k)} e_{vw}\right) = \sum_{w \in N(v)} \text{message}_w^{(k)} \odot \frac{\exp\left(f_{NN}^{(e_{vw})}\left(h_w^{(k)}\right)\right)}{\sum_{w' \in N(v)} \exp\left(f_{NN}^{(e_{vw'})}\left(h_{w'}^{(k)}\right)\right)}, \tag{7}$$

where f_{NN} is a feed forward neural network and ⊙ denotes the Hadamard product. We use Gate Recurrent Unit (GRU) [41] as update function. GRU is the best update function in original MPAN [30]. We compare the performance of the model with different update functions. We show the result in Supplementary Table 1, see Supplementary Data available online at http://bib.oxfordjournals.org/

$$h_v^{(k+1)} = \text{GRU}\left(h_v^{(k)}, m_v^{(k)}\right). \tag{8}$$

After K times of message passing, hidden features of each node contain the messages of its Kth neighbor nodes.

Readout

We use the residual idea to connect the hidden feature$h_v^{(K)}$with the initial feature $h_v^{(0)}$ as the new node feature. The purpose of using residual model is to propagate the input signal directly from the lower layer to the higher layer during the forward propagation, which adds more information to the output and thus enhances the robustness of the model. Then, the features of all nodes are aggregated into the representation of graph. Since each atom has different contributions to the physicochemical properties of the drug, we calculate the attention score as the weight coefficient of the atom. The formula for this part is as following:

$$T_{\mathrm{mpan}}^{(d_i)} = \mathrm{Readout}\left(h_v^{(K)}, h_v^{(0)} \,|v \in G_i\right) \tag{9}$$

$$\begin{aligned}&\mathrm{Readout}\left(h_v^{(K)}, h_v^{(0)}\right)\\&= \sum_{v\in G} p_{NN}\left(\left(h_v^{(K)}, h_v^{(0)}\right)\right) \odot \frac{\exp\left(g_{NN}\left(\left(h_v^{(K)}, h_v^{(0)}\right)\right)\right)}{\sum_{w'\in G}\exp\left(g_{NN}\left(\left(h_{w'}^{(K)}, h_{w'}^{(0)}\right)\right)\right)},\end{aligned} \tag{10}$$

[[DmEquation10]]where p_{NN} and g_{NN} are feed forward neural network, ⊙ denotes Hadamard product and (,) denotes connecting operation.

We use MPAN on each SMILES string to generate the feature vector $T_{\mathrm{mpan}}^{(d_i)}$, which contains information on both atoms, chemical bonds and graph structure of the drug. Compared with the traditional Graph Convolutional Network (GCN), MPAN pays attention to both neighbor nodes and connected edges between nodes when updating node features. Therefore, the features generated by MPAN can adequately represent the 2D structural information of drugs.

As shown in Figure 2, at each time step, each node shares information with its neighbors and updates its hidden features. As the time step increases, the hidden features of node h_v^k capture a broader view of its local constraint environment, which is represented by gray atoms. And after K times message passing, the hidden features of each atom are updated K times and we get an atomic feature matrix $N \in R^{n\times r}$.

Sequence feature encoder: transformer

Transformer, proposed in [31], relies on attention mechanism to calculate contextual features, which is obviously different from Recurrent Neural Network (RNN) [42] and CNN [40]. Transformer is suitable for encoding sequential information and it is widely used in NLP. Multihead attention mechanism enables Transformer to learn the features of different subsequences in the sequence. Transformer is capable of linking different positions of a sequence to obtain an embedding containing contextual information when processing sequence information.

We input the FCS-encoded sequence into Transformer to generate a feature vector $T_{\mathrm{transformer}}^{(d_i)}$, which contains 1D sequence structure information.

$$S_i = \mathrm{FCS}\left(d_i\right) \tag{11}$$

$$T_{\mathrm{transformer}}^{(d_i)} = \mathrm{Transformer}\left(S_i\right). \tag{12}$$

Multidimensional feature decoder

This module contains two parts: first is feature hybrid part, where we hybridize the features obtained from the 2D graph feature encoder and the 1D sequence feature encoder. We apply two hybrid methods, sum and concatenation, which are two traditional feature hybrid methods that have been widely used in previous studies [20, 22, 24, 27]. Previous studies have shown that the use of these two hybrid methods helps the model achieve its goal. We obtain two different matrices representing drug features from Graph encoder and Sequence encoder, respectively. We have two methods to combine the features, by summing or concatenating the two matrixes.

For the ith drug, denoted by d_i, $T_{\mathrm{mpan}}^{(d_i)}$ and $T_{\mathrm{transformer}}^{(d_i)}$ are two features obtained from the 2D graph feature encoder and the 1D sequence feature encoder. We can sum and concatenate the feature matrixes through the following formulas (13) and (14):

$$\mathrm{sum}\left(T_{\mathrm{mpan}}^{(d_i)}, T_{\mathrm{transformer}}^{(d_i)}\right) = T_{\mathrm{mpan}}^{(d_i)} + T_{\mathrm{transformer}}^{(d_i)} \tag{13}$$

$$\mathrm{concat}\left(T_{\mathrm{mpan}}^{(d_i)}, T_{\mathrm{transformer}}^{(d_i)}\right), = \left(T_{\mathrm{mpan}}^{(d_i)}, T_{\mathrm{transformer}}^{(d_i)}\right) \tag{14}$$

where '+' denotes vector addition and (,) denotes connecting operation.

In predicting interactions, the multidimensional features of two drugs are fed into a decoder, which eventually outputs a final token indicating DDIs $\hat{P} \in 0$ or 1. We use a decoder consisting of a three-layer feedforward neural network capable of establishing a strong association between input features and output results.

The loss function

In order to establish a DDIs prediction model, we construct a two-channel multidimensional drug feature encoder. MPAN is applied to 2D graph feature encoder, and Transformer is applied to 1D sequence feature encoder. Multidimensional feature decoder decodes all embedding vectors from the embedding space. The output of the decoder is a result $\hat{P} \in 0, 1$, where 1 indicates that a DDI is predicted to present and 0 indicates that there is no DDI predicted. Our model is optimized using a cross-entropy loss function, where Prepresents the true label

$$\mathrm{Loss} = -\sum P \log \hat{P} - \lambda\left(1 - P\right)\log\left(1 - \hat{P}\right). \tag{15}$$

Backpropagation spread from the output layer to each previous layer. We train the model with all trainable parameters by this end-to-end approach. The results show that end-to-end training can greatly improve the performance of the model, since all trainable parameters accept the gradient of the loss function. In this study, loss is propagated through two-channel multidimensional drug feature encoder and multidimensional feature decoder.

Data experiments

Data set and setup

Data set

The DDIs data are obtained from the Drugbank [43]. We get 24 516 positive samples from Drugbank; each sample contains a pair of drugs and a label meaning whether they can interact with each other (label 0 means no interaction with each other and vice versa). We get SMILES of all drugs from Drugbank where we observed the number of atoms on all drugs vary over a wide range (between 1 and 781). Thus, we first remove unqualified SMILES along with their associated samples if drug SMILES have atom numbers larger than 50, and the remaining number of positive samples is 17 866. Second of all, we discard drug SMILES that cannot be successfully converted into graph by RDKit [32] and delete their associated samples at the time, and the remaining number of positive sample is 17 141. Finally, negative samples are generated by randomly pairing drugs as along as this drug pair does not have any known DDIs in DrugBank. We balance the ratio of positive and negative samples to 1:1. Eventually, 34 282 DDI relationships with 17 141 positive DDI samples and 17 141 negative samples involving 1537 drugs were obtained.

The data content used within the data set is: (i) The compound ID of drug 1; (ii) The SMILES of drug 1; (iii) The compound ID of drug 2; (iv) The SMILES of drug 2 and (v) DDI label.

Baseline methods

To test the performance of our method, we compare AMDE with the following baseline methods. Random forest (RFC) [44] is a widely used classification method dealing with sequence data; One-Dimension Convolutional Neural Networks (1D-CNNs), Seq2seq [18], Long Short-Term Memory (LSTM) [45] and Deepconv [46] are neural networks for operating on sequence data. Graph Attention Network (GAT) [47], DeepDrug [22] and Knowledge Graph Neural network (KGNN) [24] are classification models based on graph.

(i) RFC is a machine learning classification algorithm. The Morgan fingerprint pairs of drugs are concatenated as a high-dimensional vector and fed into a classifier for prediction outcome. It can handle thousand-dimensions of input vector without dimension deletion and runs efficiently on large data sets [44].

(ii) 1D-CNN is a neural network for processing sequential data. It uses an integrated fingerprint of Morgan fingerprint and circular fingerprint as input and predicts DDIs by using CNN [40].

(iii) DeepDrug is a deep learning method for predicting DDIs. It uses SMILES string as input and four-layers GCN to extract the features of the drug. Pairs of drug features are sent to a fully connected layer to predict the DDI outcome [22].

(iv) KGNN is a knowledge graph-based DDI prediction method. A knowledge graph is constructed for all drugs in which features of drugs in the knowledge graph are encoded by GCN and the obtained drug feature vectors are fed into a dense layer to predict DDI outcome [24].

(v) Seq2seq [18] is an algorithm based on RNN [42]. Morgan fingerprint is used as input; then, an RNN is used as encoder to encode drug feature, and eventually, another RNN is used as decoder to predict whether DDI exists.

(vi) Deepconv [46] uses Morgan fingerprint as input. It extracts drug features through DNN. The features of drug pairs are concatenated as input to a dense layer to predict DDI outcome.

(vii) LSTM [45] is a method of processing time series. It uses Morgan fingerprint as input. After embedding Morgan fingerprint, LSTM is used to encode features. Pairs of drug features are sent to a fully connected layer to predict the DDI outcome.

(viii) DeepDDI [48] is a deep learning method for predicting DDIs. It constructs a drug similarity matrix based on fingerprint and then reduces the feature dimension using PCA and predicts DDI through DNN.

(ix) GAT [47] is a graph-based deep learning algorithm. It uses one-layer GCN with eight attention heads to extract drug features, then the drug pair features are sent to a feed forward neural network to predict DDI outcome.

Metrics

We denote the true label and predicted values of DDIs by P and $\hat{P}$, respectively. Four metrics are applied to evaluate the performance of the model, including accuracy (ACC), area under ROC curve (AUROC), area under PRC curve (PRAUC) and F1 score. These metrics have different emphasis. ACC focuses on assessing the model's ability to correctly classify samples, while F1 focuses on assessing the model's sensitivity. When dealing with classification problems, AUROC is suitable for class-balanced data sets, while PRAUC is better able to distinguish the generalization ability of models in the case of unbalanced data sets.

Evaluation settings

Different validation methods have been used for measuring the model performance. We refer to previous studies [34, 49] using two types of data sets, 100 and 50%.

Initially, we randomly select half of the data from 100% data set to form 50% data set. There are 34 282 samples in 100% data set and 17 141 samples in 50% data set, and they both involved 1537 drugs. Here, we divided the both data set into training set, validation set and test set in the ratio of 8:1:1. For each model, we repeated the experiment five times to obtain reliable and stable results on 100 and 50% data sets, respectively. For each data experiment, we keep the same values of the involved

Table 1. Performance of AMDE-sum and AMDE-cat in different message passing number (MP-Num) (AMDE-sum and AMDE-cat mean the use of different hybrid method in our model). The best result is in bold for each evaluation metric. F1 represents F1 score, ACC represents accuracy, AUROC represents area under ROC curve

MP-Num	AMDE-cat			AMDE-sum		
	F1	ACC	AUROC	F1	ACC	AUROC
1	97.53 ± 0.12	97.33 ± 0.18	98.97 ± 0.22	97.55 ± 0.12	97.24 ± 0.13	98.90 ± 0.13
2	97.60 ± 0.13	97.63 ± 0.22	99.01 ± 0.09	97.33 ± 0.25	96.68 ± 0.21	98.90 ± 0.23
3	97.57 ± 0.24	97.43 ± 0.21	98.37 ± 0.13	97.37 ± 0.17	96.42 ± 0.18	98.04 ± 0.19
4	97.22 ± 0.27	97.63 ± 0.19	98.03 ± 0.22	97.53 ± 0.30	96.44 ± 0.17	98.11 ± 0.22
5	96.57 ± 0.19	97.54 ± 0.13	97.90 ± 0.17	97.40 ± 0.18	96.19 ± 0.22	98.54 ± 0.16
6	95.24 ± 0.17	97.36 ± 0.12	98.15 ± 0.21	96.33 ± 0.22	96.15 ± 0.18	98.11 ± 0.17

parameters and use the same training set, validation set and test set.

AMDE is implemented on PyTorch1.0.1 [50]. For graph encoder, we set the size of the message vector to 25, the times of message passing of two hybrid methods sum and concatenation are 1 and 2 (we analyze this parameter in the **Sensitivity analysis** section) and the size of the generated 2D graph feature vector to 75. For sequence encoder, we set the number of attention heads of Transformer to 8 and the length of the final feature to 75. We set the batch size to 128 and allow AMDE to run for 50 epochs. We use Adam Optimizer with learning rate of 1e−4 to select the best performing model from the validation set based on AUROC performance, and the model selected by validation is evaluated in the test set.

Sensitivity analysis

For choosing the hyperparameters that make our model have the best performance, we do some sensitivity analysis experiments. We discuss the changes of various performance parameters of the model in different message passing number (MP-Num). We present the performance variation of the model at MP-Num = {1, 2, 3, 4, 5, 6} in Table 1. More detailed experimental results are shown in supplementary Table 2.

It is found that performance of our model has very small fluctuation as the value of message passing number increases. This means that too many times of message passing cannot contribute more valuable information. Under the two feature hybrid methods, AMDE-sum and AMDE-cat all perform well. The highest average F1, ACC and AUROC of AMDE-sum are 97.60, 97.63 and 99.01%. The highest average F1, ACC and AUROC of AMDE-cat are 97.55, 97.24 and 98.90%. It is found that the standard deviation displacement of the AMDE is quite small under all MP-Num, which indicates that the framework of our model has a certain stability. The value of parameters is the ones with the best performance: MP-Num = 1 in AMDE-sum and MP-Num = 2 in AMDE-cat (AMDE-sum and AMDE-cat mean the hybrid method).

Results and analysis

To test the validity and robustness of AMDE, we investigate AMDE-sum and AMDE-cat with sum or concatenation in feature hybrid. AMDE-avg represents the average performance of AMDE-sum and AMDE-cat. We compare them with the baseline models on Drugbank [43] data set. We would like to test the performance of our model with less training data sets. Therefore, we randomly select a half of the data from 100% data set to form 50% data set (there are 34 282 samples in 100% data set, 17 141 samples in 50% data set and they both involved 1537 drugs).

Data experiments are repeated five times on 100 and 50% data sets, respectively. The mean and standard deviation of all metrics are reported in Table 2. AMDE achieves higher mean and small standard deviation in all metrics, which achieve the best performance. It is used four-layers GCN in DeepDrug to extract atom graph feature of drugs. Pairs of features are concatenated and passed to a dense layer to compute the final prediction. Deepdrug performs better than GAT, which may be due to the more layers of GCN used in DeepDrug because those more layers make the vision of nodes wider and the information contained in node richer. This shows that the feature encoder based on atomic graph should pay attention to the multihop neighbor information, which is essential to the model performance for the DDIs prediction. Encoding multihop neighbor information requires more resources.

In AMDE, the graph encoder encodes the edge information between connected atoms when encoding node features, which adds more information to node features and improves the quality of DDIs prediction tasks. The performance of AMDE is also better than some models which encode the sequence feature of drugs from fingerprints (RFC, 1D-CNN, Seq2seq, LSTM, Deepconv and DeepDDI). It is necessary to encode sequence features and atomic graph features simultaneously for DDIs prediction task. KGNN regards drug entities and entities in other fields as nodes and the relationship between entities as edges. From the perspective of KGNN, the feature of each drug is on node level. The drug features encoded by our AMDE are on graph level. The performance of KGNN is worse than AMDE, indicating that viewing drugs as nodes in an interaction graph is improper.

We conduct experiments at 50% data set to observe the dependence on the amount of data of all models. It is worth noticing that when we reduce the data by

Table 2. Comparison between AMDE and baseline models. Bold indicates the highest evaluation metric scores, and '–' means not available. AMDE-sum and AMDE-cat mean the use of different hybrid method in our model; AMDE-avg is the average performance of AMDE-sum and AMDE-cat. F1 represents F1 score, ACC represents accuracy and AUROC represents area under ROC curve; 100% data set means 34 282 samples, and 50% data set means 17 141 samples

Model	**100% data set**			**50% data set**		
	F1	**ACC**	**AUROC**	**F1**	**ACC**	**AUROC**
RFC	78.33 ± 0.28	78.90 ± 0.33	–	83.45 ± 0.22	80.64 ± 0.28	–
DeepDDI	81.43 ± 0.51	90.44 ± 0.43	92.18 ± 0.47	80.15 ± 0.40	89.54 ± 0.63	90.18 ± 0.33
1-D CNN	95.45 ± 0.50	95.33 ± 0.45	97.45 ± 0.18	94.22 ± 0.44	94.03 ± 0.31	97.62 ± 0.20
GAT	95.87 ± 0.19	95.39 ± 0.21	97.30 ± 0.18	95.80 ± 0.17	95.33 ± 0.20	96.88 ± 0.19
Seq2Seq	96.55 ± 0.21	96.01 ± 0.20	98.41 ± 0.19	95.37 ± 0.12	95.25 ± 0.25	97.86 ± 0.21
LSTM	96.35 ± 0.18	96.14 ± 0.22	98.49 ± 0.12	95.92 ± 0.16	95.72 ± 0.21	98.07 ± 0.18
DeepDrug	96.62 ± 0.22	95.89 ± 0.21	98.50 ± 0.16	97.17 ± 0.18	95.62 ± 0.17	98.51 ± 0.21
Deepconv	97.19 ± 0.19	97.30 ± 0.17	98.61 ± 0.22	97.23 ± 0.15	96.89 ± 0.12	98.47 ± 0.20
KGNN	96.30 ± 0.25	97.51 ± 0.13	98.67 ± 0.09	94.55 ± 0.20	94.23 ± 0.14	97.60 ± 0.17
AMDE-sum	97.55 ± 0.12	97.24 ± 0.13	98.90 ± 0.13	97.63 ± 0.11	97.64 ± 0.13	99.41 ± 0.14
AMDE-cat	97.60 ± 0.13	97.63 ± 0.22	99.01 ± 0.09	98.11 ± 0.09	97.37 ± 0.17	99.50 ± 0.11
AMDE-avg	97.57±0.13	97.43 ± 017	98.95 ± 0.11	97.87 ± 0.10	97.50 ± 0.15	99.45 ± 0.18

half, AMDE still maintains a stable performance and all evaluation metrics remain optimal. We show the comparison of all metrics when the model is trained using 100 and 50% data sets, respectively, in Figure 3 (the results of DeepDDI and RFC are not shown in Figure 3 for better observation).

The evaluation metrics of our model remain stable when the data set is reduced, which demonstrates the good robustness of our model. The experiment results show that our model AMDE is able to learn the characteristic patterns of drugs from less data, which overcomes the difficulties caused by small data sets to some extent. The baseline model (RFC, 1D-CNN, Seq2seq, LSTM, Deepconv, DeepDDI, DeepDrug, GAT and KGNN) only learns single dimensional features. When the data set is reduced, the features learned by these models become inaccurate. AMDE learns the 1D sequence features and 2D graph features from SMILES strings, which encodes the information of the drug more comprehensively. AMDE encodes features from multiple dimensions, which can learn more features than other methods even with less data for training. As a result, AMDE shows consistent performance on data sets of different volumes (100 and 50%) in DDIs predictions.

Simple ablation study

As mentioned earlier, the existing DDIs prediction models have limitations as they only learn the features of drug from a single dimension. In this section, we consider verifying the significance of our proposed model AMDE to encode features in each dimension and investigate the impact of simultaneous use of multidimensional features on DDI prediction. The experiment is repeated five times on each encoder to obtain reliable and stable results on 100 and 50% data sets, respectively. The mean and standard deviation of all metrics are reported in Table 3.

It is found that our AMDE method outperforms the single-dimensional feature encoder in all evaluation metrics. Meanwhile, the fluctuation range of standard deviation of AMDE in all evaluation metrics is also very small. It shows that a feature encoder considering multiple dimensions can adequately encode the features of drugs, therefore improving the effect of DDIs prediction task.

Identify potential DDIs

In this section, we discuss the ability of AMDE to predict the potential DDIs. Firstly, it is selected a set of DDIs that have never appeared in the training, validation and test sets before. The data set contains 3723 samples involving 150 drugs (1675 positive and 2048 negative samples). Since the data set is unbalanced, we add an evaluation metric PRAUC, which can better evaluate the generalization ability of the model on unbalanced data set. We use the trained models to predict these samples. The mean and standard deviation are shown in Table 4. Figure 4 shows the ROC curve and PRC curve of compare models. KGNN, DeepDrug and Deepconv are selected as the baseline, which perform well in the comparative experiment. Our model AMDE shows better and stable performance in the new DDIs prediction task.

Visual analysis

T-distributed stochastic neighbor embedding (t-SNE) [51] is a machine learning algorithm used for dimensionality reduction, which can visualize high-dimensional data, so that we have an intuitive understanding of the distribution of data. To further investigate why our model is so effective, we reduce the dimension of the embedding vectors learned by attention-based multidimensional feature encoder (AMDE) to three dimensions by *t*-SNE method for visualization, as shown in Figure 5. The embedding vectors learned by our proposed model can

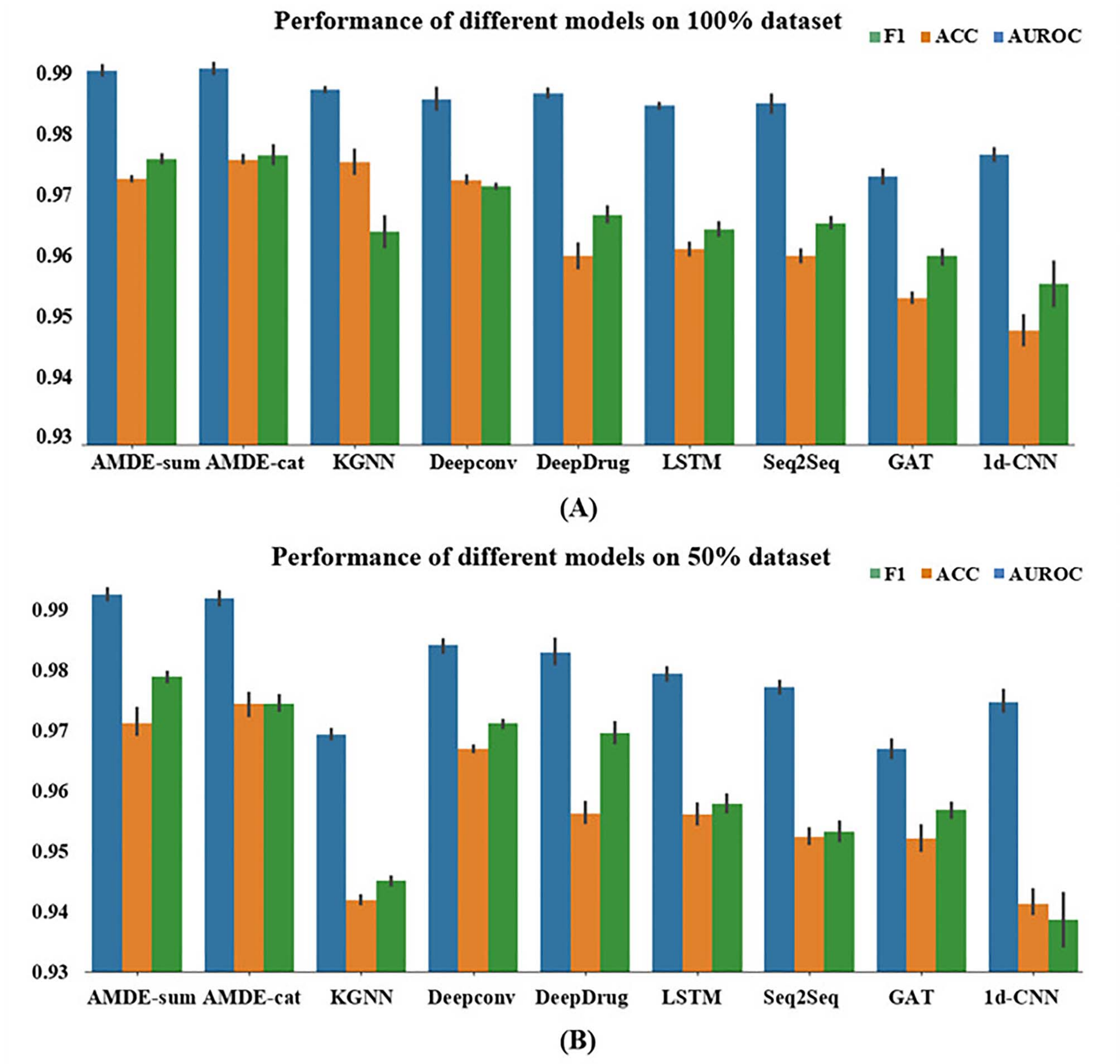

Figure 3. Performance of different models on 100% data set (**A**) and 50% data set (**B**).

Table 3. The experimental results of ablation of AMDE. Transformer represents sequence encoder, and MPAN represents graph encoder. The best result is in bold for each evaluation metric. F1 represents F1 score, ACC represents accuracy and AUROC represents area under ROC curve

Model	**100% data set**			**50% data set**		
	F1	**ACC**	**AUROC**	**F1**	**ACC**	**AUROC**
Transformer	97.03 ± 0.19	96.86 ± 0.12	98.52 ± 0.16	95.03 ± 0.21	95.91 ± 0.17	97.47 ± 0.12
MPAN	97.26 ± 0.16	97.14 ± 0.17	98.61 ± 0.13	97.41 ± 0.11	97.14 ± 0.14	98.42 ± 0.17
AMDE-sum	97.55 ± 0.12	97.24 ± 0.13	98.90 ± 0.13	97.63 ± 0.11	97.64 ± 0.13	99.41 ± 0.14
AMDE-cat	97.60 ± 0.13	97.63 ± 0.22	99.01 ± 0.09	98.11 ± 0.09	97.37 ± 0.17	99.50 ± 0.11

easily separate interacting drug pairs from noninteracting drug pairs. This means that our model can learn more differentiated feature representations, which is the key for our model to perform well in the DDIs prediction task.

Case analysis

Torasemide (DB00214) was initially used as a potent diuretic, which is later found to be able to control blood pressure and treat edema caused by heart failure, kidney disease, cirrhosis of the liver [52], etc. It has been well

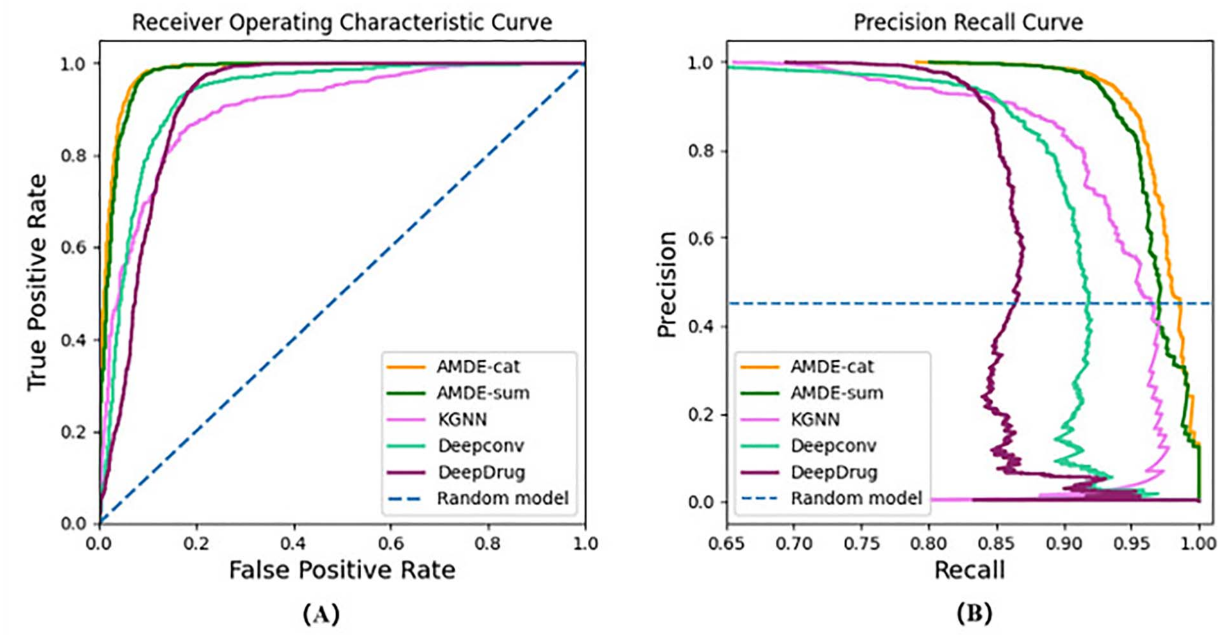

Figure 4. Receiver Operating Characteristic Curve (**A**) and Precision Recall Curve (**B**) of compare models for new DDIs predict. The dotted line represents a completely random model and the ratio of positive cases in the data set, for (**A** and **B**), respectively.

Table 4. Experimental results for compare models for new DDIs predict. The best result is in bold for each evaluation metric. AMDE-sum and AMDE-cat mean the use of different hybrid method in our model. ACC represents accuracy, AUROC represents area under ROC curve and PRAUC represent area under PRC curve

Model	ACC	AUROC	PRAUC
KGNN	81.05 ± 0.34	89.97 ± 0.51	93.33 ± 0.23
DeepDrug	87.17 ± 0.22	91.28 ± 0.18	85.72 ± 0.14
Deepconv	87.79 ± 0.23	92.81 ± 0.19	89.47 ± 0.21
AMDE-sum	92.81 ± 0.21	97.57 ± 0.20	96.82 ± 0.17
AMDE-cat	90.03 ± 0.27	98.14 ± 0.19	**97.5 0**±0.13

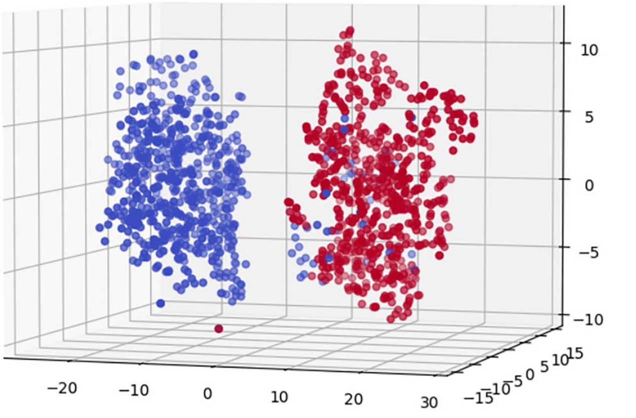

Figure 5. *t*-SNE feature dimension reduction. The features extracted by our AMDE are downscaled by using *t*-SNE and reduced to 3D space. Features can be clearly distinguished. Where red indicates drug pairs predicted to have interaction and blue indicates drug pairs predicted not to have interaction.

tolerated with adverse effects of a mild, transient nature reported by only small numbers of patients. Therefore, it is valuable to further explore the function of Torasemide (DB00214) in combination [53]. We use Torasemide (DB00214) as an example and predict the most likely drug to interact with it using our proposed model. The drugs with the top 15 scores are shown in Table 5, all of which were confirmed to interact with Torasemide (DB00214). For drugs that do not appear in training process, the results show that our model AMDE can also accurately predict drugs with potential interactions with them.

Conclusion

In this paper, we proposed a DDIs prediction model AMDE, which encodes the multidimensional features of drugs from the atomic graph and subsequences of the drug at the same time. Extensive experiments have shown that our model can achieve state-of-the-art prediction performance, demonstrating the power of multidimensional features in DDI prediction tasks. Our model is a promising framework, and future work can further optimize our model. It can also be applied to other problems such as drug-target prediction, cancer risk prediction and so on.

Table 5. Top 15 drugs predicted for Torasemide (DB00214)

Rank	Drugbank ID	Drug name	Verification method
1	DB00458	Imipramine	Drugbank
2	DB00363	Clozapine	Drugbank
3	DB01104	Sertraline	Drugbank
4	DB01242	Clomipramine	Drugbank
5	DB01224	Quetiapine	Drugbank
6	DB01267	Paliperidone	Drugbank
7	DB00726	Trimipramine	Drugbank
8	DB00476	Duloxetine	Drugbank
9	DB00176	Fluvoxamine	Drugbank
10	DB01175	Escitalopram	[54]
11	DB00334	Olanzapine	Drugbank
12	DB00679	Thioridazine	Drugbank
13	DB01238	Aripiprazole	Drugbank
14	DB01095	Fluvastatin	Drugbank
15	DB00682	Warfarin	Drugbank

Materials and methods

The code is available at https://github.com/wan-Ying-Z/AMDE-master.

Data availability statement

The DruBank dataset is open-source.

Key Points

- We propose an attention-based multidimensional feature encoder (AMDE), which encodes one-dimensional sequence features and two-dimensional atomic graph features of drugs. The features it encodes represent the information of drugs more precisely.
- The multidimensional feature decoder further compresses the drug feature vector and is able to strongly associate features with prediction results.
- It is provided a new method of feature fusion: integrating drug features from multiple dimensions can enhance the effectiveness of downstream prediction tasks.
- AMDE has achieved advanced results in predicting new DDIs.

Supplementary data

Supplementary data are available online at https://academic.oup.com/bib.

Acknowledgements

We thank our partners who provided all the help during the research process and the team for their great support.

Funding

Natural Science Foundation of China (grant nos 61873280, 61972416, 61873281); Taishan Scholarship (grant no. tsqn201812029); Natural Science Foundation of Shandong Province (grant no. 2019GGX101067); Foundation of Science and Technology Development of Jinan (grant no. 201907116); Fundamental Research Funds for the Central Universities (grant no. 18CX02152A, 19CX05003A-6); Intelligent Diagnosis of Early Lung Cancer Classification Based on Self-learning Pulsed Neural Membrane System and Its Clinical Application (grant no. 05E19070003); 3D Reconstruction and Accurate Volume Measurement of White Matter High Signal Based on Deep Learning and Its Clinical Application (grant no. 05tc2107001); Research on Topological Representation and ultra-low complexity computability of large-scale graph data (grant no. 05E18070010); Juan de la Cierva and Talento-Comunidad de Madrid.

References

1. Rockey WM, Elcock AH. Progress toward virtual screening for drug side effects. *Proteins Struct Funct Genet* 2002;**48**(4):664–71.
2. Goettler M, Schneeweiss S, Hasford J. Adverse drug reaction monitoring – cost and benefit considerations. *Pharmacoepidemiol Drug Saf* 2015;**6**(S3):S79–90.
3. Kansal S, Bansod PP, Kumar A. Prediction of instantaneous heart rate using adaptive algorithms. *Int J Adap Inno Syst* 2019;**2**:267.
4. Rashdan S, Yang H, Le T, *et al.* Prevalence and significance of potential pharmacokinetic drug–drug interactions among patients with lung cancer: implications for clinical trials. *Clin Drug Investig* 2021;**41**(2):161–7.
5. Sacan A, Ekins S, Kortagere S. Applications and limitations of in silico models in drug discovery. *Methods Mol Biol* 2012;**910**:87–124.
6. Gowri R, Kanmani S. Self-adaptive agent-based tutoring system. *Int J Adap Inno Syst* 2015;**2**:197.
7. Tao S, Xun W, Xin L, *et al.* A programming triangular DNA origami for doxorubicin loading and delivering to target ovarian cancer cells. *Oncotarget* 2017.
8. Wang S, Liu J, Ding M, *et al.* DL-SMILES#: a novel encoding scheme for predicting compound protein affinity by deep learning. *Comb Chem High Throughput Screen* 2021;**24**.
9. Vilar S, Harpaz R, Uriarte E, *et al.* Drug-drug interaction through molecular structure similarity analysis. *J Am Med Inform Assoc* 2012;**19**(6):1066–74.
10. Vilar S, Uriarte E, Santana L, *et al.* Detection of drug-drug interactions by modeling interaction profile fingerprints. *PLoS One* 2013;**8**(3):e58321.
11. Li P, Huang C, Fu Y, *et al.* Large-scale exploration and analysis of drug combinations. *Bioinformatics* 2015;**31**:2007–16.
12. Vilar S, Uriarte E, Santana L, *et al.* Similarity-based modeling in large-scale prediction of drug-drug interactions. *Nat Protoc* 2014;**9**:2147–63.
13. Zhang W, Chen Y, Liu F, *et al.* Predicting potential drug-drug interactions by integrating chemical,biological, phenotypic and network data. *BMC Bioinform* 2017;**18**:18.
14. Zhang P, Wang F, Hu J, *et al.* Label propagation prediction of drug-drug interactions based on clinical side effects. *Sci Rep* 2015;**5**:12339.
15. Liu S, Zhang Y, Cui Y, *et al.* Enhancing drug-drug interaction prediction using deep attention neural networks. *bioRxiv* 2021.
16. Deng Y, Xinran X, Qiu Y, *et al.* A multimodal deep learning framework for predicting drug–drug interaction events. *Bioinformatics* 2020;**36**(15):4316–22.

17. Mikolov T, Chen K, Corrado G, *et al.* Efficient estimation of word representations in vector space. *Comput Sci* 2013.
18. Xu Z, Wang S, Zhu F, *et al.* Seq2seq fingerprint: an unsupervised deep molecular embedding for drug discovery. *ACM Int Conf Bioinform ACM* 2017.
19. Deac A, Huang Y-H, Veličković P, *et al. Drug-Drug Adverse Effect Prediction with Graph Co-Attention*. 2019. ArXiv:1905.00534.
20. Xin CA, Xl A, Ji W. GCN-BMP: investigating graph representation learning for DDI prediction task. *Methods* 2020;**179**:47–54.
21. Cho H, Choi IS. Enhanced deep-learning prediction of molecular properties via augmentation of bond topology. *ChemMed Chem* 2019;**14**(17):1604–9.
22. Cao X, Fan R, Zeng W. DeepDrug: a general graph-based deep learning framework for drug relation prediction. *Biorxiv* 2020.
23. Wang X, Liu D, Zhu J, *et al.* CSConv2d: a 2-D structural convolution neural network with a channel and spatial attention mechanism for protein-ligand binding affinity prediction. *Biomolecules* 2021;**11**(5):643.
24. Lin X, Quan Z, Wang ZJ, *et al.* KGNN: knowledge graph neural network for drug-drug interaction prediction. In: *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI-20)*. 2020.
25. Wang S, Du Z, Ding M, *et al.* KG-DTI: a knowledge graph based deep learning method for drug-target interaction predictions and Alzheimer's disease drug repositions. *Appl Intell* 2021;**4**.
26. Yu Y, Huang K, Zhang C, *et al.* SumGNN: multi-typed drug interaction prediction via efficient knowledge graph summarization. In: *International Conference on Intelligent Systems for Molecular Biology Proceedings*, 2020;**37**(18):2988–95.
27. Xu H, Sang S, Lu H. *Tri-Graph Information Propagation for Polypharmacy Side Effect Prediction*. NeurIPS, 2019.
28. Bang S, Jong HJ, Hyunjung S. Polypharmacy side-effect prediction with enhanced interpretability based on graph feature attention network. *Bioinformatics* 2021;**37**(18):2955–62.
29. Marinka Z, Monica A, Jure L. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 2018;**13**: 457–66.
30. Withnall M, Lindelf E, Engkvist O, *et al.* Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. *J Chem* 2020;**12**(1):1.
31. Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. *NIPS* 2017.
32. https://www.rdkit.org/docs/.
33. Wishart DS. Introduction to cheminformatics. *Curr Protoc Bioinformatics* 2016;**53**(1).
34. Huang K, Xiao C, Glass L, *et al.* MolTrans: molecular interaction transformer for drug target interaction prediction. *Bioinformatics* 2020;**37**(6):830–36.
35. Gilmer J, Schoenholz SS, Riley PF, *et al. Neural message passing for quantum chemistry*. ArXiv:170401212 Cs, June 14, 2017, preprint: not peer reviewed.
36. Jo J, Kwak B, Choi HS, *et al.* The message passing neural networks for chemical property prediction on SMILES. *Methods* 2020;**179**: 65–72.
37. Wang Z, Liu M, Luo Y, *et al.* Advanced graph and sequence neural networks for molecular property prediction and drug discovery. 2021. ArXiv: 2012.01981.
38. Datta R, Das D, Das S. Efficient lipophilicity prediction of molecules employing deep-learning models. *Chemom Intell Lab Syst* 2021;**213**:104309.
39. http://www.rdkit.org/docs/GettingStartedInPython.html.
40. Lecun Y, Bengio Y. Convolutional networks for images, speech, and time-series. *Handbook of Brain Theory & Neural Networks*, 1995.
41. Chung J, Gulcehre C, Cho K, *et al.* Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv: Neural and Evolutionary Computing*, 2014.
42. Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncertainty Fuzziness Knowl-Based Syst* 1998;**6**(2):107–16.
43. Wishart DS, Craig K, Guo AC, *et al.* DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 2008;**36**:D901–6.
44. Breiman L. Random forests. *Mach Learn* 2001;**45**:5–32.
45. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;**9**(8):1735–80.
46. Lee I, Keum J, Nam H. DeepConv-DTI: prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Comput Biol* 2019;**15**(6):e1007129.
47. Velikovi P, Cucurull G, Casanova A, *et al.* Graph attention networks. 2018. ArXiv:1710.10903.
48. Ryu JY, Kim HU, Sang YL. Deep learning improves prediction of drug–drug and drug–food interactions. *Proc Natl Acad Sci U S A* 2018;**115**(18):E4304.
49. Shukla PK, Shukla PK, Sharma P, *et al.* Efficient prediction of drug-drug interaction using deep learning models. *IET Syst Biol* 2020;**14**(4):211–6.
50. Paszke A, Gross S, Chintala S, *et al. Automatic differentiation in PyTorch*. NIPS, 2017.
51. Maaten LvD, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**(Nov):2579–605.
52. Young M, Plosker GL. Torasemide. *PharmacoEconomics* 2001;**19**(6): 679–703.
53. Friedel HA, Buckley MT. Torasemide. *Drugs* 1991;**41**(1):81–103.
54. Rosner MH. Severe hyponatremia associated with the combined use of thiazide diuretics and selective serotonin reuptake inhibitors. *Am J Med Sci* 2004;**327**(2):109–11.