Graph Transformer for Drug Response Prediction

Thang Chu, Thuy Trang Nguyen, Bui Duong Hai, Quang Huy Nguyen, and Tuan Nguyen

Abstract—*Background*: Previous models have shown that learning drug features from their graph representation is more efficient than learning from their strings or numeric representations. Furthermore, integrating multi-omics data of cell lines increases the performance of drug response prediction. However, these models have shown drawbacks in extracting drug features from graph representation and incorporating redundancy information from multi-omics data. This paper proposes a deep learning model, GraTransDRP, to better drug representation and reduce information redundancy. First, the Graph transformer was utilized to extract the drug representation more efficiently. Next, Convolutional neural networks were used to learn the mutation, meth, and transcriptomics features. However, the dimension of transcriptomics features was up to 17737. Therefore, KernelPCA was applied to transcriptomics features to reduce the dimension and transform them into a dense presentation before putting them through the CNN model. Finally, drug and omics features were combined to predict a response value by a fully connected network. Experimental results show that our model outperforms some state-of-the-art methods, including GraphDRP and GraOmicDRP.

Index Terms—Deep learning, drug response prediction, graph transformer, graph convolutional network, kernel PCA

1 Introduction

Personalized medicine is a rapidly advancing field in finding the specific treatment best suited for an individual based on their biological characteristics. Its approach relies on the understanding of an individual's molecular and genomics profile [1]. However, conducting drug trials on each individual to observe corresponding responses is expensive and not ethical. These challenges continue to be problematic for large-scale studies on this topic [2]. Therefore, cell lines of tumor samples have been obtained and developed as "artificial patients" to study the drug response.

Recently, projects such as GDSC [3], CCLE [4] and NCI60 [5] have published a database containing different types of omics data, such as copy number aberration, gene expression, methylation, etc. These data help to facilitate the development of computational methods for drug response prediction [6], [7], [8], [9], [10], [11], [12]. As a result, competitions (Dream challenge) have been opened to direct the attention of researchers, scientists, and scholars toward having this problem solved [13]. From simple to complex models, various machine learning algorithms have been applied to solve the drug sensitivity problem, namely support vector machines (SVMs), linear regression, and neural network models [14][15] [16]. More advanced methods such as multiple-kernel, multiple-task learning, and collaborative filtering techniques were proposed to integrate various types of -omics data or different individual models to boost the performance [17] [18], [19], [20]. Meanwhile, graph-based methods focusing on biological perspective have been introduced, including structural similarities between drugs, biological similarities between cell lines, interactions between proteins and gene regulatory [14], [21].

The machine learning-based methods above have proven their ability through the accuracy of drug response prediction. However, limitations still exist, like drugs and cell lines' representation. High dimensional data often represents them since each -omics profile can contain thousands of genes for each cell line. Similarly, there are many chemical and structural features for each drug. Due to the limited number of cell lines, current machine learning methods have to face "small n, large p" or "The curse of dimensionality" problem. As the number of dimensions increases, the volume of our domain increases exponentially, which requires more samples so that the model can learn efficiently. As a result, traditional machine learning-based methods will likely become underfitting, and their predictive ability will be decreased.

Recently, deep learning as a branch of machine learning has become increasingly popular. It can learn a complex data representation in the higher dimension and make a more accurate prediction than traditional machine learning-based methods. [22]. Due to its usefulness, deep learning has been applied to facilitate the development of computational biology. Specifically, it outperforms traditional machine learning-based algorithms in numerous drug repositioning, visual screening, and drug-target profiling [23], [24], [25], [26], [27], [28], [29], [30]. By reducing the noise, deep learning helps to extract a better drug representation and other biological data. [31], [32].

For drug response problems, deep learning is utilized to automatically learn genomic features and transcriptomic and epigenomic features of cell lines. Also, it can extract the chemical structures of drugs to predict the sensitivity of anticancer drugs. Therefore, deep learning does not need to calculate molecular features or perform feature selections, which is susceptible to errors. Various models have been proposed to address this issue. [33], [34], [35], [36]. For

- *Thang Chu is with the Faculty of Science, University of Alberta, Edmonton, AB T6G 2R3, Canada. E-mail: chuducthang77@gmail.com.*
- *Thuy Trang Nguyen, Bui Duong Hai, Quang Huy Nguyen, and Tuan Nguyen are with the Faculty of Mathematical Economics, National Economics University, Hanoi 100000, Vietnam. E-mail: {thuytrang, haibd, huynqtkt, nttuan}@neu.edu.vn.*

Manuscript received 8 December 2021; revised 6 June 2022; accepted 6 September 2022. Date of publication 15 September 2022; date of current version 3 April 2023.
(Corresponding author: Tuan Nguyen.)
Digital Object Identifier no. 10.1109/TCBB.2022.3206888

instance, DeepDR, tCNNS, and CDRScan can be considered some of the original works in the field. Using a deep neural network, DeepDR predicts the half-maximal inhibitory concentrations (IC50) by building a multiple sub-network to learn the drug representation. Besides, tCNNS uses a convolutional neural network to extract the features of drugs from SMILES string representations. However, since drug and cell line data contain redundant information, other models apply dimension reduction techniques such as autoencoder and variational autoencoder to solve this problem. Specifically, DeepDSC [35] uses a deep autoencoder to extract genomics features of cell lines from gene expression data and then combine them with chemical features of compounds to predict drug responses. Other methods have been experimented with to solve this problem from a different perspective, such as MOLI, which is a drug-specific model, but shares the same approach of using a convolutional neural network as DeepDR [37]. These deep learning models generally use either strings or numerical drugs, which are not natural data representations. Therefore, the structural information of drugs may be lost. As a result, some current methods have represented the drug structure as a graph and used graph convolutional networks to learn drug features.

Graph convolutional networks (GCN) have been applied to learn the representations of compound structures depicted as molecular graphs [38] [39]. GraphDRP [38] outperforms other string-representation-based models such as tCNNs by using GCN to represent the drugs' graph where the edges are the bonding of atoms. There are also studies suggesting that these omics data are more informative than the genomic data of cell lines [40]. Therefore, GraomicDRP [39] was recently published and shown to be a state-of-the-art method among other deep learning-based methods in drug response prediction tasks. GraomicDRP is built based on the GraphDRP model. However, instead of using only genomic data, GraomicDRP combines different types of -omics data, namely transcriptomic data (i.e., gene expression) and epigenomic data (i.e., methylation), to increase the performance. As a result, our work will be compared directly with GraomicDRP.

In this study, we propose GraTransDRP (Graph Transformer for drug response prediction), a novel neural network architecture capable of extracting a better drug representation from molecular graphs to predict drug response on cell lines. Graph Transformer was integrated with the combination of GAT-GCN to enhance the ability to predict more accurate drug responses. We also incorporated the idea of using multi-omics data from GraomicDRP. Finally, we compared our method with GraphDRP and GraomicDRP in both single-omics, and multi-omics data [38], [39]. The experimental results indicate that our method performs better in Root Mean Square Error (RMSE) and Pearson correlation coefficient for all experiments. Furthermore, while both genomic and epigenomic data are presented in a binary format with 377 and 735 dimensions, respectively, the transcriptomic data is continuous and normalized in the range from 0 to 1 with 17737 dimensions. Therefore, Kernel PCA was applied to reduce the dimension of transcriptomic data into the same dimension as genomic and epigenomic data. Then our experiment indicates that the combination between Graph Transformer and GAT-GCN using all the processed -omics data achieves the highest result among all the possible combinations.

2 Graph Transformer for Drug Response Prediction (GraTransDRP)

This study proposed the model GraTransDRP, which takes chemical information of drugs and multi-omics data of cell lines to predict response values. The proposed model is shown in Fig. 1.

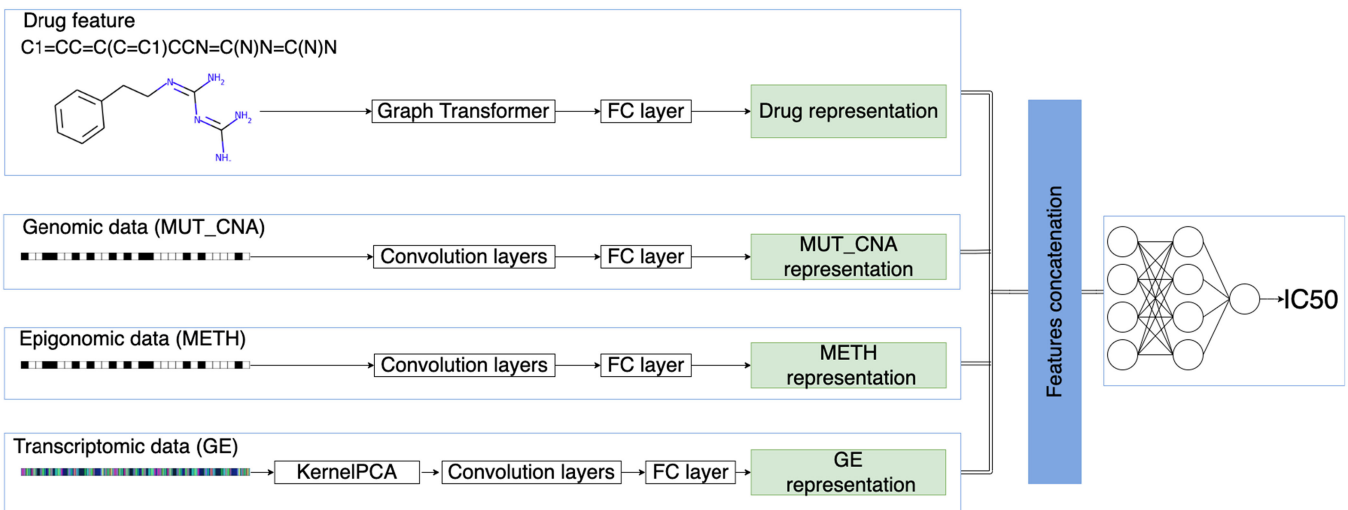

Fig. 1. An illustration of GraTransDRP. Genomic data and epigenomic data were converted to one-hot format with a vector of 735 and 337 dimensions, respectively. Initially, gene expression data was put through KernelPCA to reduce dimension. Then 1D convolutional layers were applied three times to these features. After that, the fully connected (FC) layer was used to convert ConvNet results into 128 dimensions. The drug in the SMILE string was converted to graph format. Then Graph Transformer was used to learn the drug's features. Following the Graph Transformer, the fully connected layer was also used to convert the result to 128 dimensions. Finally, three feature representations and drug representation were then concatenated and put through two FC layers to predict the IC50 value.

For drug features, the drugs represented in SMILES format [41] were downloaded from PubChem [42]. Then, RDKit, an open-source chemical informatics software [43], was used to construct a molecular graph reflecting the interactions between the atoms inside the drug. Atom feature design from DeepChem [44] was used to describe a node in the graph. Each node contains five atom features: atom symbol, atom degree calculated by the number of bonded neighbors and Hydrogen, the total number of Hydrogen, implicit value of the atom, and whether the atom is aromatic. These atom features constituted a multi-dimensional binary feature vector [45]. If there exists a bond among a pair of atoms, an edge is set. As a result, an indirect, binary graph with attributed nodes was built for each input SMILES string. Graph Transformer was integrated with GAT-GCN model [45] to learn the features of drugs. Following the graph neural network, a fully connected layer (FC layer) was also used to convert the result to 128 dimensions.

The genomic and epigenomic features of cell lines were represented in one-hot encoding. As for, KernalPCA was applied to reduce the dimension from 17737 to 1000. Then, 1D convolutional neural network (CNN) layers were used to learn latent features on those data. The output of each feature was put through a fully connected layer to output a 128 dimension vector of cell line representation. Finally, in the 512-dimension vector, the combination of drugs' features and cell lines' features were put through two fully-connected layers, in which the number of nodes are 1024 and 256, respectively, before the response was predicted.

2.1 Graph Convolutional Networks (GCN)

Formally, a graph for a given drug $G = (V, E)$ was stored in the form of two matrices, including feature matrix X and adjacency matrix A. X $\in \mathbb{R}^{N\times F}$ consisted of N nodes in the graph, and each node was represented by F-dimensional vector. $A \in \mathbb{R}^{N\times N}$ displayed the edge connection between nodes. The original graph convolutional layer took two matrices as input and aimed to produce a node-level output with C features for each node. The layer was defined as:

$$AXW, \tag{1}$$

where $W \in \mathbb{R}^{F\times C}$ was a trainable parameter matrix. However, there were two main drawbacks. First, for every node, all feature vectors of all neighboring nodes were summed up, but not the node itself. Second, matrix A was not normalized, so the multiplication with A would change the scale of the feature vector. GCN model [46] was introduced to solve these limitations by adding an identity matrix to A and normalizing A. Also, it was found that symmetric normalization achieved more interesting results. The GCN layer was defined by [46] as:

$$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW, \tag{2}$$

where $\tilde{A}$ is the graph adjacency matrix with added self-loop, $\tilde{D}$ is the graph diagonal degree matrix.

2.2 Graph Attention Networks (GAT)

The self-attention technique has been shown to be self-sufficient for state-of-the-art level results on machine translation [47]. Inspired by this idea, the self-attention technique was used in the graph convolutional network in GAT [48]. We adopted a graph attention network (GAT) in our model. The proposed GAT architecture was built by stacking a *graph attention layer*. The GAT layer took the node feature vector **x**, as input, then applied a linear transformation to every node by a weight matrix **W**. Then the *attention coefficients* were computed at every pair of nodes where the edge exists. For example, the coefficients between node i and j were computed as:

$$a(\mathbf{W}\mathbf{x}_i, \mathbf{W}\mathbf{x}_j). \tag{3}$$

This value indicates the importance of node j to node i. These *attention coefficients* were then normalized by applying a soft-max function. Finally, the output features for each node were computed as

$$\sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\mathbf{W}\mathbf{x}_j\right), \tag{4}$$

where $\sigma(.)$ is a non-linear activation function and α_{ij} are the normalized *attention coefficients*.

2.3 Combined Graph Neural Network (GAT&GCN)

A combination of GAT [48] and GCN [46] was also proposed to learn graph features [45]. At first, the GAT layers learned to combine nodes in an attention manner, so after the GAT layers, these features in each node were abstract, containing high-level information about the graph. Finally, GCN layers were used to learn convolved features and combine these abstract nodes to make final predictions.

2.4 Graph Transformer

Recently, Graph Attention Network (GAT) and Graph Convolution Network (GCN) have been used to learn the drug representation. These techniques are designed to learn on homogeneous graphs, while a drug's graph can be a heterogeneous graph containing different node and link types. Also, missing connections among nodes in the graph should be taken into account when the features are extracted. Graph Transformer could learn better features for a more generalized drug graph by considering these missing links. Transformer techniques have been well-applied in a natural language processing to solve the bottlenecks of Recurrent Neural Network (RNN). It uses multi-head attention to allow a word to attend to each other terms in a sentence. Inspired by this idea, Graph Transformer allows for the con-tribute of neighborhood nodes in extracting graph features. Formally, a heterogeneous graph $G = (V, E)$ has a set of node types, $\mathcal{T}^v$, and a set of edge type $\mathcal{T}^e$. Compactly, there is an adjacency tensor $\mathbb{A} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}\times\mathrm{K}}$, where $K = |\mathcal{T}^e|$, and feature matrix $X \in \mathbb{R}^{N\times F}$ consists of N nodes in the graph and each node is represented by F-dimensional vector. Also, a meta-path to predict new connections among nodes is defined as

$$A_{\mathcal{P}} = A_{t1}\ldots A_{tp}, \tag{5}$$

where A_{ti} is an adjacency matrix for the ith edge type of meta-path. For each adjacency matrix, a soft adjacency

matrix Q using 1x1 convolution is chosen as

$$Q = F(A, W_\phi) = \phi(A, \mathrm{softmax}(W_\phi)), \tag{6}$$

where ϕ is a convolution layer and $W_\phi \in R^{1\times1\times K}$. Multiply these soft adjacency matrices Q together, and a convex combination of new meta-paths is established. Combining with the Graph Convolution network, node representations are formed as

$$Z = ||_{i=1}^{C} \sigma(D_i^{-1}\hat{A}_i^{(l)} XW). \tag{7}$$

The above equation is a function of neighborhood connectivity, which is similar to the attention mechanism shown in the transformer for natural language processing. To enhance the prediction accuracy, Graph Transformer considers the positional encoding, which encodes the distance-aware information. In contrast to the natural language processing problem, the position of nodes is difficult to determine when extracting features due to the natural properties of the graph. Therefore, Graph Transformer uses Laplacian eigenvectors to address this issue. These eigenvectors are computed as

$$\Delta = I - D^{-1/2}AD^{-1/2} = U^T \Lambda U, \tag{8}$$

where U and Λ are eigenvectors and eigenvalues, respectively. Grah Transformer uses Laplacian eigenvectors to pre-compute all the graphs in the dataset. In our model, we used two layers of Graph Transformer with the combination of GAT-GCN in the middle to enhance the ability to extract features and increase the prediction's accuracy.

2.5 Kernel PCA

Kernel PCA is an extension of PCA [49] using kernel methods. Besides the curse, there is also a bless of dimensionality so that N points can almost always be linearly separable. While PCA is a linear method, Kernel PCA deals with a non-linear situation where it uses the kernel function to project the dataset into a higher dimensional space. Instead of working a the high dimensional space, N-by-N kernel dimensional space is created as

$$k_{x,y} = k(\boldsymbol{\Phi(x), \Phi(y)}) = \Phi(\boldsymbol{x})^T \Phi(\boldsymbol{y}), \tag{9}$$

where Φ can be chosen arbitrarily, and each element in the matrix represents the similarity of one transformed data with respect to all the transformed data. Then, the same procedure as PCA on this matrix is applied. Among various kernel functions, the radial basis function (RBF) is the most well-known kernel. With two samples x and x', the kernel of the two samples is calculated as:

$$K(\boldsymbol{x}, \boldsymbol{x'}) = exp(-\frac{||\boldsymbol{x} - \boldsymbol{x'}||}{2\sigma^2}), \tag{10}$$

where σ is the width of the Gaussian distribution. The resulting feature is bounded between zero and one. It is close to one if x is similar to x' and close to zero if two data points are dissimilar. Our study chose RBF as the kernel function to reduce the dimension of gene expression data to combine with methylation and mutation data to predict drug response.

3 Experimental Setting

3.1 Datasets

CCLE [4] and GDSC [3] are extensive drug sensitivity screening projects containing not only –omics but also drug response data for anti-cancer drugs on thousands of cell lines. The –omics data includes gene expression (i.e., transcriptomic data), which indicates several RNAs transcribed from DNA and thus the amount of translated proteins in a cell. Therefore, the expression level of a gene indicates the activity level of a gene in a particular state (e.g., diseased or normal) in a cell. In addition, the –omics data also implies genomic aberrations such as mutations and copy number variations (CNVs) in genome. In this study, we also used methylation (epigenomic data), which regulates gene expression by recruiting proteins involved in gene repression or inhibiting transcription factors' binding to DNA. Meanwhile, cell lines were cultured and treated with different doses of drugs. Therefore, we used IC50 as the drug response measure of efficiency to inhibit the vitality of cancer cells. Specifically, it indicates the amount of a particular drug needed to inhibit the biological activity by half.

GDSC is the most extensive database containing a diversity of -omics data and cell line information. In a previous study on GraomicDRP [39], there were 223 drugs tested on 948 cell lines in that database. Meanwhile, only 24 drugs were tested on 504 cell lines in CCLE. Thus, we selected GDSC for this study. Similarly, we have three types of -omics data, including Gene expression (GE), mutation and copy number aberration (MUT_CNA), and methylation (METH). The detail of the dataset is shown in Table 1. In addition, a cell line was described by a binary vector, where 1 or 0 indicates whether or not a cell line has a genomic aberration, respectively. Methylation data is also in binary form to show whether a gene is hyper-methylated or hypo-methylated. Gene expression data provides a continuous expression level for genes, which is normalized between 0 and 1. The response values in terms of IC50 were also normalized in a range (0,1) as in [50]. Meanwhile, drugs were represented in canonical SMILES format [41].

3.2 Experimental Design

In this section, the performance of our model is demonstrated through three experiments: the integration of Graph Transformer with single, multi-omics data, the effect of dimensional reduction on gene expression data,

TABLE 1
Dataset

Dataset		#Cell lines	#Known drug-cell line responses
Single -omic	METH	676	150,761
	GE	857	191,034
	MUT_CNA	857	191,049
multi- omic	METH & GE	663	147,891
	GE & MUT_CNA	838	186,864
	MUT_CNA & METH	676	150,761
	ALL	663	147,891

and blind drugs/cell-lines test. Because the GraomicDRP model uses the same techniques as GraphDRP, they only combined multi-omics data to achieve better performance. So in the single-omic experiment, we compared the model's performance with GraphDRP, and in the multi-omics experiment, we compared the model's performance with GraomicDRP.

Of all cell line pairs from the GDSC database, we split it into 80% as the training set, 10% as the validation, and 10% as the test set. The validation set is used to tune the hyperparameters, and the testing set is used to evaluate the model's generalization. Initially, we chose the values of the hyperparameters based on the previous works. Then we tuned many parameters such as learning rate and batch size to achieve a better result. Detailed experiments are described below.

3.2.1 The Integration of Graph Transformer

This experiment aims to test the effect of the Graph Transformer model with all combinations of -omics data. Previously, GraomicDRP tested single and multi-omics data to observe which combination achieved the highest result. In this experiment, the Graph Transformer was combined with GAT-GCN and evaluated on the same procedure as before. First, the model's performance was recorded when testing with single-omics data. Then, the model was evaluated with a combination of two and all-omics data, respectively. Finally, we compared the results to discover which variety of -omics data has the highest prediction accuracy.

3.2.2 The Effect of Dimensional Reduction on Gene Expression Data

This experiment aims to study the effect of different dimensional reduction techniques on gene expression data to obtain the overall result. Observing the dataset, we noticed that the dimension of gene expression data is significantly larger than the dimension of mutation and methylation data. Therefore, it potentially contains redundant information and incurs noise in our prediction. We investigated this effect using dimensional reduction techniques on gene expression features, then compared the performance with original -omics data and the best result in the first experiment. Specifically, PCA, KernelPCA, and Isomap were used to reduce gene expression data.

3.2.3 Blind Drugs/Cell-Lines Test

In the previous experiment, a drug/cell line that appeared in the testing set may also appear in the training phase, causing the over-estimation of the generalization of the model. Therefore, this experiment was designed to evaluate the prediction performance of unseen drugs/cell lines.

Initially, the experiment was designed to test unseen cell lines. Therefore, the dataset was split to guarantee that cell lines in the training dataset are different from cell lines in the test dataset. A total of 90% (891/990) cell lines were randomly selected, and their IC50 values were kept for the training phase. The remaining 10% (99/990) cell lines were used as the testing set.

Similarly, it is sometimes required to make predictions for a new drug not in the training phase. So we also experimented with testing the prediction of unseen drugs. Drugs were constrained from existing in training and testing at the same time. Of 90% (201/223) drugs, their IC50 values were randomly selected for train- ing, including 80% drugs for the training set and 10% drugs for the validation set. 10% (22/223) drugs were used as the testing set in the remaining set.

TABLE 2
Performance Comparison in Terms of CC_p and RMSE on the GDSC Dataset With the Integration of Single-Omics Data

Method	GraTransDRP		GraphDRP	
	CC_p	RMSE	CC_p	RMSE
METH	0.9218	0.0251	0.9104	0.0279
GE	0.9279	0.0249	**0.9165**	**0.0259**
MUT_CNA	**0.9317**	**0.0238**	0.9120	0.0263

The best performance is in Bold.

3.3 Performance Evaluation

Root mean square error (RMSE) and Pearson correlation coefficient (CC_p) are adopted to evaluate the performance of models. RMSE measures the difference between the predicted value and the true value:

$$RMSE = \sqrt{\frac{1}{n}\sum_i^n (o_i - y_i)^2}, \tag{11}$$

where n is the number of data points, o_i is a ground-truth of ith sample, y_i is the predicted value of ith sample.

Pearson correlation coefficient measures how strong the relationship is between two variables. CC_p is defined as:

$$CC_p = \frac{\sum_i^n (o_i - y_i)^2}{(\sigma_O \sigma_Y)} \tag{12}$$

where the standard deviation of O and Y is σ_O, σ_Y respectively. The lower RMSE, the better the model is. Meanwhile, the higher CC_p, the better the model is.

4 Results and Discussion

4.1 The Integration of Graph Transformer

Tables 2 and 3 present the prediction performance in terms of CC_p and RMSE for different experiments by the baseline (GraomicDRP [39]) and our proposed method (GraTransDRP).

Tables 2 and 3 show the prediction performance of the models using single and multi-omics data. We observed that our proposed models, for all kinds of -omics data, achieved better RMSE and CC_p than GraomicDRP. The Graph Transformer is more efficient than Graph Convolutional Network in extracting drug features.

The GraomicDRP's experiment showed that gene expression data has the best performance in terms of RMSE (0.0259) and CC_p (0.9165) for single-omics data. Meanwhile, our model suggested that mutation and copy number aberration data has the highest performance (0.9317 for CC_p and 0.0238 for RMSE). Within each type of -omics data, we can observe that GraTransDRP outperforms GraomicDRP for CC_p and RMSE. For multi-omics data, while the combination of methylation and gene expression achieves the highest performance in the GraomicDRP model, GraTransDRP indicates

TABLE 3
Performance Comparison in Terms of CC_p and RMSE on the GDSC Dataset With the Integration of Multi-Omics Data

Method	GraTransDRP		GraomicDRP	
	CC_p	RMSE	CC_p	RMSE
METH & GE	0.9351	0.0236	**0.9310**	**0.0239**
GE & MUT_CNA	0.9301	0.0337	0.9236	0.0246
MUT_CNA & METH	**0.9353**	**0.0235**	0.9277	0.0252
ALL	0.9341	0.0239	0.9295	0.0244

The best performance is in Bold.

that the combination of mutation and methylation achieves a higher result. Specifically, the CC_p result of MUT_CNA & METH in GraTransDRP is 0.9353, higher than the CC_p result of METH & GE in GraomicDRP (0.9310). Similarly, we observed that the performance of GraTransDRP outperforms the performance of GraomicDRP in all combinations.

However, both models point out that combining all three -omics data does not necessarily have the best performance. We suspected this due to redundant information introduced by new types of -omics data. Therefore, we experimented with different dimensional reduction techniques on gene expression data to investigate whether the dimensionality reduction techniques improved the result or not.

Fig. 3 shows ten drugs that achieved the highest and lowest performance when GE & METH (Fig. 2A) are combined in terms of RMSE and CCp.

4.2 The Effect of Dimensional Reduction on Gene Expression Data

In this experiment, we applied techniques for dimensionality reduction, namely PCA [49], Isomap [51], and KernelPCA [52]. PCA can be viewed as a linear method, while KernelPCA and Isomap are better for non-linearity cases. We first applied these techniques to gene expression and then combined all three -omics data in the final model. Table 4 shows that applying dimensional reduction techniques improves the previous performance (0.9341) in terms of CC_p. Furthermore, we observed that KernelPCA achieved the best performance among three-dimensional reduction techniques, with 0.9356 for CC_p and 0.0235 for RMSE. This experiment suggests that the gene expression data is not linearly separable in the current dimension. Instead, KernelPCA and Isomap methods project the current data into the higher dimensional space, which exists a linearly separable hyperplane through the dataset. Since Kernel PCA has a higher experimental result, we focused on analyzing this technique.

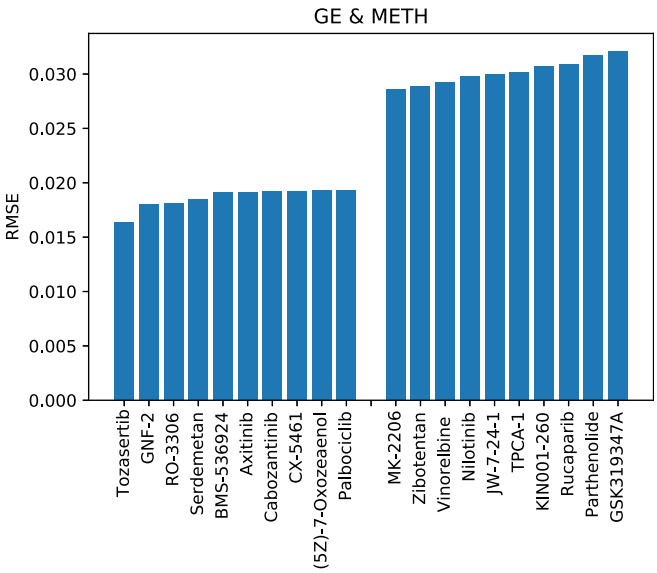

Fig. 2. Ten drugs obtained the highest (on the left) and lowest (on the right) performance in the combination of GE & METH achieved best performance in terms of RMSE.

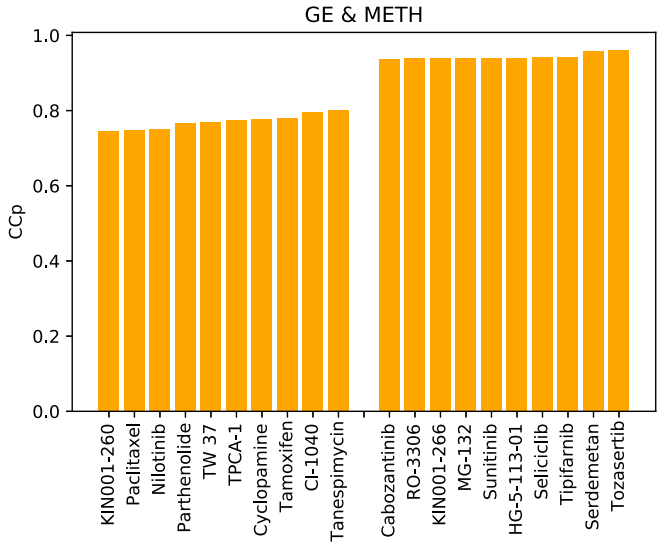

Fig. 3. Ten drugs obtained the highest (on the left) and lowest (on the right) performance in the combination of GE & METH achieved best performance in terms of CCp.

Graph Transformer using all -omics data with KernelPCA technique achieves the best prediction performance in terms of RMSE and CC_p among all experiments in this study. It confirms our suspicion that integrating more features is not necessarily good in terms of performance as there is a redundancy in -omic data. As a result, applying dimensional reduction techniques to -omics data before the model improves the performance. Also, it unleashes the potential of Graph Transformer in graph representation, partly supporting the claim in [53] that Graph Transformer extracts better node features in the heterogeneous graph.

4.3 Blind Drugs/Cell-Lines Test

This experiment evaluates the performance of the model on unseen drugs/cell-lines. Drugs/cell-lines are prevented from existing in the training and testing phase at the same

TABLE 4
Performance Comparison in Terms of CC_p and RMSE on the GDSC Dataset With the Integration of Multi-Omics Data

Method		CC_p	RMSE
GraTransDRP		0.9341	0.0239
GraTransDRP	KernelPCA	**0.9356**	**0.0235**
	PCA	0.9342	0.0239
	Isomap	0.9343	0.0238

The best performance is in Bold.

TABLE 5
Performance Comparison in Terms of CC_p and RMSE on the GDSC Dataset in the Blind Test With Unseen Drug and Cell-Line Experiment

Methods	Unseen Drug		Unseen Cell-line	
	CC_p	RMSE	CC_p	RMSE
GraOmicDRP	0.4309	0.0590	0.8766	0.0327
GraTransDRP	**0.4401**	**0.0499**	**0.9027**	**0.0314**

The best performance is in bold.

time. For response prediction of unknown drugs/cell lines, Table 5 shows our models, GraTransDRP outperformed GraOmicDRP in terms of both RMSE and CCp.

However, the performance of our model in a blind experiment on drugs/cell lines is significantly lower than the result in the above experiments. This is because this experiment is more challenging. Specifically, the drugs/cell lines in the test set do not appear in the training set. This indicates that it is harder to predict the drug-cell line response for unseen drugs or unseen cell-lines.

5 Conclusions and Discussion

In this article, we proposed a novel method for drug response prediction called GraTransDRP. Our model used Graph Transformer, similar to the well-known Transformer technique in natural language processing, to extract a better drug representation. The original GAT-GCN model is combined with Graph Transformer to enhance the features of drugs and the overall drug response prediction. Instead of using only single-omics data as in GraphDRP, we also tested our model on multi-omics data. Using the 1D convolutional neural network, we extracted the cell line features of multi-omics data before combining them with drug representation. Also, since the dimension is significantly larger than the other two -omics data, KernelPCA was applied to this data to reduce the noise it incorporates into the final prediction. Finally, all features were combined to predict the IC50 value.

The performance of our proposed method was compared against the state-of-the-art drug response prediction models, including GraphDRP and GraomicDRP. Because the GraomicDRP model uses the same techniques as GraphDRP, they only combined multi-omics data to achieve better performance. So in the single-omic experiment, we compared the model's performance with GraphDRP, and in the multi-omics experiment, we compared it with GraomicDRP. Also, we designed a blind drug and blind cell-lines experiment so that the drug and cell line appearing in the training set will not be present in the testing set. Therefore, we reduced the over-estimation of the generalization of the model. The experimental results indicated that our method achieves better performance in both Root Mean Square Error and Pearson correlation coefficient. The performance suggests that graph transformer extracts drug features better than Graph Convolution Networks.

Acknowledgments

Availability of data and materials: https://github.com/chuducthang77/GraTransDRP.

References

[1] I. S. Chan and G. S. Ginsburg, "Personalized medicine: Progress and promise," *Annu. Rev. Genomic. Hum. Genet.*, vol. 12, no. 1, pp. 217–244, 2011, [Online]. Available: https://doi.org/10.1146/annurev-genom-082410–101446

[2] J. N. Weinstein et al., "The cancer genome atlas pan-cancer analysis project," *Nature News*, Sep. 2013. [Online]. Available: https://www.nature.com/articles/ng.2764

[3] W. Yang et al., "Genomics of drug sensitivity in cancer (GDSC): A resource for therapeutic biomarker discovery in cancer cells," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D955–D961, 2012.

[4] J. Barretina et al., "The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity," *Nature*, vol. 483, no. 7391, pp. 603–607, 2012.

[5] R. H. Shoemaker, "The NCI60 human tumour cell line anticancer drug screen," *Nature Rev. Cancer*, vol. 6, no. 10, pp. 813–823, 2006.

[6] F. Azuaje, "Computational models for predicting drug responses in cancer research," *Brief. Bioinf.*, vol. 18, no. 5, pp. 820–829, 2017.

[7] J. Chen and L. Zhang, "A survey and systematic assessment of computational methods for drug response prediction," *Brief. Bioinf.*, 2020. [Online]. Available: https://doi.org/10.1093/bib/bbz164

[8] M. J. Garnett et al., "Systematic identification of genomic markers of drug sensitivity in cancer cells," *Nature*, vol. 483, no. 7391, pp. 570–575, 2012.

[9] I. S. Jang, E. C. Neto, J. Guinney, S. H. Friend, and A. A. Margolin, "Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data," in *Biocomputing*. Singapore: World Scientific, 2014, pp. 63–74.

[10] H. Liu, Y. Zhao, L. Zhang, and X. Chen, "Anti-cancer drug response prediction using neighbor-based collaborative filtering with global effect removal," *Mol. Ther. Nucleic Acids*, vol. 13, pp. 303–311, 2018.

[11] L. Zhang, X. Chen, N.-N. Guan, H. Liu, and J.-Q. Li, "A hybrid interpolation weighted collaborative filtering method for anti-cancer drug response prediction," *Front. Pharmacol.*, vol. 9, 2018, Art. no. 1017.

[12] N.-N. Guan, Y. Zhao, C.-C. Wang, J.-Q. Li, X. Chen, and X. Piao, "Anticancer drug response prediction in cell lines using weighted graph regularized matrix factorization," *Mol. Ther. Nucleic Acids*, vol. 17, pp. 164–174, 2019.

[13] J. C. Costello et al., "A community effort to assess and improve drug sensitivity prediction algorithms," *Nature Biotechnol.*, vol. 32, no. 12, 2014, Art. no. 1202.

[14] T. Turki and Z. Wei, "A link prediction approach to cancer drug sensitivity prediction," *BMC Syst. Biol.*, vol. 11, no. 5, 2017, Art. no. 94.

[15] A. Seal and D. J. Wild, "Netpredictor: R and shiny package to perform drug-target network analysis and prediction of missing links," *BMC Bioinf.*, vol. 19, no. 1, 2018, Art. no. 265.

[16] C. Huang, R. Mezencev, J. F. McDonald, and F. Vannberg, "Open source machine-learning algorithms for the prediction of optimal cancer drug therapies," *PLoS One*, vol. 12, no. 10, 2017, Art. no. e0186906.

[17] M. Gönen and A. A. Margolin, "Drug susceptibility prediction against a panel of drugs using kernelized bayesian multitask learning," *Bioinformatics*, vol. 30, no. 17, pp. i556–i563, 2014.

[18] K. Matlock, C. De Niz, R. Rahman, S. Ghosh, and R. Pal, "Investigation of model stacking for drug sensitivity prediction," *BMC Bioinf.*, vol. 19, no. 3, 2018, Art. no. 71.

[19] M. Tan, O. F. Özgül, B. Bardak, I. Ekşioğlu, and S. Sabuncuoğlu, "Drug response prediction by ensemble learning and drug-induced gene expression signatures," *Genomics*, vol. 111, no. 5, pp. 1078–1088, 2019.

[20] Q. Wan and R. Pal, "An ensemble based top performing approach for NCI-DREAM drug sensitivity prediction challenge," *PLoS One*, vol. 9, no. 6, 2014, Art. no. e101183.

[21] N. Zhang, H. Wang, Y. Fang, J. Wang, X. Zheng, and X. S. Liu, "Predicting anticancer drug responses using a dual-layer integrated cell line-drug network model," *PLoS Comput. Biol.*, vol. 11, no. 9, 2015, Art. no. e1004498.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[23] A. Gonczarek, J. M. Tomczak, S. Zarba, J. Kaczmar, P. Dbrowski, and M. J. Walczak, "Interaction prediction in structure-based virtual screening using deep learning," *Comput. Biol. Med.*, vol. 100, pp. 253–258, 2018.

[24] J. C. Pereira, E. R. Caffarena, and C. N. dos Santos, "Boosting docking-based virtual screening with deep learning," *J. Chem. Inf. Model.*, vol. 56, no. 12, pp. 2495–2506, 2016.
[25] M. Karimi, D. Wu, Z. Wang, and Y. Shen, "DeepAffinity: Interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks,," *Bioinformatics*, vol. 35, no. 18, pp. 3329–3338, 2019.
[26] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: Deep drug–target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.
[27] L. Wang et al., "A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network," *J. Comput. Biol.*, vol. 25, no. 3, pp. 361–373, 2018.
[28] M. Wen et al., "Deep-learning-based drug–target interaction prediction," *J. Proteome Res.*, vol. 16, no. 4, pp. 1401–1409, 2017.
[29] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov, "Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data," *Mol. Pharmaceutics*, vol. 13, no. 7, pp. 2524–2530, 2016.
[30] X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, and F. Cheng, "deepDR: A network-based deep learning approach to in silico drug repositioning," *Bioinformatics*, vol. 35, no. 24, pp. 5191–5198, 2019.
[31] I. I. Baskin, D. Winkler, and I. V. Tetko, "A renaissance of neural networks in drug discovery," *Expert Opin. Drug Discov.*, vol. 11, no. 8, pp. 785–795, 2016.
[32] A. Lavecchia, "Deep learning in drug discovery: Opportunities, challenges and future prospects," *Drug Discov. Today*, vol. 24, pp. 2017–2032, 2019.
[33] Y. Chang et al., "Cancer drug response profile scan (CDRscan): A deep learning model that predicts drug effectiveness from cancer genomic signature," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, 2018.
[34] Y.-C. Chiu et al., "Predicting drug response of tumors from integrated genomic profiles by deep neural networks," *BMC Med. Genomic.*, vol. 12, no. 1, 2019, Art. no. 18.
[35] M. Li et al., "DeepDSC: A deep learning method to predict drug sensitivity of cancer cell lines," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, pp. 575–582, 2019.
[36] P. Liu, H. Li, S. Li, and K.-S. Leung, "Improving prediction of phenotypic drug response on cancer cell lines using deep convolutional network," *BMC Bioinf.*, vol. 20, no. 1, 2019, Art. no. 408.
[37] X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, and F. Cheng, "deepDR: A network-based deep learning approach to in silico drug repositioning," *Bioinformatics*, vol. 35, no. 24, pp. 5191–5198, 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz418
[38] T.-T. Nguyen, G. T. T. Nguyen, T. Nguyen, and D.-H. Le, "Graph convolutional networks for drug response prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, pp. 146–154, 2021.
[39] G. T. T. Nguyen, D.-H. Vu, and D.-H. Le, "Integrating molecular graph data of drugs and multiple -omic data of cell lines for drug response prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, pp. 710–717, 2022.
[40] F. Iorio et al., "A landscape of pharmacogenomic interactions in cancer," *Cell*, vol. 166, no. 3, pp. 740–754, 2016.
[41] N. M. O'Boyle, "Towards a universal SMILES representation-a standard method to generate canonical SMILES based on the InChI," *J. Cheminformatics*, vol. 4, no. 1, 2012, Art. no. 22.
[42] S. Kim et al., "PubChem substance and compound databases," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D1202–D1213, 2016.
[43] G. Landrum, "RDKit: Open-source cheminformatics," 2021. [Online]. Available: http://www.rdkit.org
[44] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing, and Z. Wu, *Deep Learning for the Life Sciences*. Sebastopol, CA, USA: O'reilly Media, 2019.
[45] T. Nguyen, H. Le, T. P. Quinn, T. Le, and S. Venkatesh, "Predicting drug–target binding affinity with graph neural networks," *bioRxiv*, 2020, Art. no. 684662.
[46] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
[47] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[48] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[49] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Comput. Geosciences*, vol. 19, no. 3, pp. 303–342, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009830049390090R
[50] M. P. Menden et al., "Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties," *PLoS One*, vol. 8, no. 4, 2013, Art. no. e61318.
[51] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[52] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998, doi: 10.1162/089976698300017467.
[53] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 32, 2019.

Thang Chu is currently working toward the MSc degree in computer science with the University of Alberta. His current research interests include application of deep learning in biology.

Thuy Trang Nguyen received the MSc degree in economic cybernetics from the National Economics University, in 2012. She is a lecturer with the Faculty of Mathematical Economics of the National Economics University. Her current research interests include statistics, probability, and econometrics and applied data science in economics and finance.

Bui Duong Hai received the master's degree from the Birmingham University, U.K., in 2003, with dissertation topic on Game theory in Monetary policy. He is lecturer with the Faculty of Mathematical Economics, National Economics University, Hanoi. His research interests include applied mathematics and statistics in economics. His publications includes books of probability, mathematical statistics, econometrics.

Quang Huy Nguyen received the PhD degree in actuarial science from the Institute of Sciences in Finance and Insurance - Claude Bernard University, in Lyon, France. He is currently the deputy head of the Faculty of Mathematical Economics, National Economics University of Hanoi in Vietnam. His research interests include applied mathematics and probabilities in economics and finance. He published a number of papers in the journal of Computational and Applied Mathematics, Journal of Statistic and risk modeling, Journal of Methodology and Computing in Applied Probability and Applied Probability Journal. In addition to his academic career, he is working as an actuarial senior expert in several life insurance companies in Vietnam.

Tuan Nguyen received the MSc degree from the University of Bristol, in 2019, in the area of machine learning. He a lecturer with National Economics University. His current research interests include topic includes application of deep learning in computer vision, natural language processing, and bioinformatic.