

# Integrating Protein-Protein Interaction Information into Drug Response Prediction by Graph Neural Encoding

Yuan Xie (✉ [xieyuan@iu.edu](mailto:xieyuan@iu.edu))

Indiana University Bloomington <https://orcid.org/0000-0003-0846-1285>

Jian Peng

University of Illinois at Urbana-Champaign

Yuan Zhou

University of Illinois at Urbana-Champaign

---

## Research article

**Keywords:** Deep learning, Graph neural networks, Drug response prediction, Protein-protein interactions

**Posted Date:** December 16th, 2019

**DOI:** <https://doi.org/10.21203/rs.2.18936/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

## RESEARCH

# Integrating Protein-Protein Interaction Information into Drug Response Prediction by Graph Neural Encoding

Yuan Xie<sup>1\*</sup>, Jian Peng<sup>2</sup> and Yuan Zhou<sup>1,3</sup>

\*Correspondence: xieyuan@iu.edu

<sup>1</sup>Computer Science Department,  
Indiana University Bloomington,  
Bloomington, USA

Full list of author information is  
available at the end of the article

## Abstract

**Background:** Predicting tumor cells' reactions to medical treatments has enormous significance for both medical and clinical studies. Many machine learning approaches have been previously applied to predict the effectiveness of anti-cancer drugs by analyzing genomic profiles. However, most of the current methods neglect the fact that proteins rarely act alone and often team up as "molecular machines" characterized by intricate dynamic connections to perform biological functions. Other studies indicate that subnetworks of protein-protein interaction (PPI) networks could be crucial for the development of cancer and could be used as new therapeutic targets. Recent advances in graphical representation learning inspired by deep neural networks have provided a potential solution to include PPI information into drug response prediction to improve performance.

**Results:** We present a graph neural network (GNN) based approach, GNDR, to incorporate PPI information into drug sensitivity prediction based on genomic profiles. By communicating information among neighborhoods on the PPI network, our model fuses genomic features into a structural PPI graph which encodes information relevant to drug responses. We compare our approach with state-of-the-art baselines on three publicly available datasets which include both genomic features and drug responses of cell lines and Patient Derived Xenografts (PDX). We investigate the impact of different types of interactions as well as the effectiveness of different graph pooling strategies. Experimental results demonstrate the superior performance of our model over the baselines. In particular, on the NIBR-PDXE dataset, our model outperforms the baselines in 21 out of 26 cancer-treatment pairs and achieves a MSE of 1.46 on the CCLE-GDSC dataset and a MSE of 0.11 on the NCI-ALMANAC dataset. In addition, our model provides substantial model interpretability in terms of gene similarity and gene importance via the graph pooling procedures.

**Conclusions:** We propose GNDR, a GNN based approach which not only possesses the large learning capacity, but provides a solution to combine PPI information with genomic features for drug response prediction. Empirical evaluation demonstrates the effectiveness of adding protein information. Our approach, as far as we know, the first GNN based model for drug response prediction, provides a promising perspective for the identification of anti-cancer therapies.

**Keywords:** Deep learning; Graph neural networks; Drug response prediction; Protein-protein interactions

## Background

Due to the heterogeneity and complexity of cancer, it remains challenging to make accurate predictions of drug responses and identify effective anti-cancer treatments. By analyzing genomic alterations and transcriptomic programming of tumors, pharmacogenomics provides a promising solution for this problem; human cell lines are genomically profiled to produce a compilation of gene expression, chromosomal copy number and massively parallel sequencing data which could be combined with computational methods to predict drug responses. Recent years have witnessed many efforts in this field. On one hand, an increasing number of datasets that include both genomic profiles and drug responses of tumors or cell lines have become available, such as the Cancer Cell Line Encyclopedia (CCLE)[1], the GDSC project [2], and the NCI-60 [3] project. On the other hand, many machine learning approaches have been applied in this field and have achieved remarkable improvements. For example, Nguyen et al. [4] exploited random forest model in predicting *in vivo* drug responses whereas Chiu et al. [5] and Xia et al. [6] employed deep neural network to predict drug responses of cell lines, which have achieved state-of-the-art results. However, all methods mentioned so far treat the genomic features in a flat fashion, neglecting the fact that proteins within a tissue rarely act alone [7] and instead they often team up as “molecular machines” characterized by intricate physicochemical dynamic connections to perform biological functions at both cellular and systems levels. PPIs play essential roles in connecting networks that relay oncogenic signals, enable the acquisition of hallmark features of cancer, and serve diverse roles in driving and maintaining the growth of cancer cells [8]. In addition, it has been experimentally validated that there exists a huge amount of different types of interactions among proteins, such as co-expression, gene-fusion, and co-occurrence and these PPI networks or subnetworks [9] have been proved to be highly related to some diseases [10] as gene–gene interactions shape complex phenotypes and modify the effects of mutations during development of disease [11]. As deep neural networks (DNN) are inspired by biological neural networks that constitute animal brains, we would hope to find a machine learning model that could utilize both the genomic features and the topological features represented by PPIs of the tumor cells to further improve the drug prediction performance.

Boosted by the recent innovations in the deep learning community, great success has been achieved in the development of GNN. Instead of convolving over fixed reception fields in a grid-like input, GNNs propagate information via the neighborhoods of nodes in an irregular graph and apply trainable functions to learn low dimensional embeddings for each node that not only encodes node features but topological information as well. In addition, by using various pooling techniques such as differential graph pooling (DIFFpool) and self-attention graph pooling (SAGpool), embeddings of each node could be aggregated to represent the entire input graph, which could be fed for many downstream tasks such as classification and regression. Besides, GNNs also enjoy the advantages of DNNs as they require no prior knowledge or assumption over data distribution and possess great capacity for learning complicated representations. This offers a natural remedy to utilize both genomic profiles and PPI information.

In this study, we propose a graph neural network based approach, GNNDR, for predicting drug responses of cell lines and xeno-grafts based on their genomic features and PPI information. Specifically, we first construct a graph of proteins connected by edges that represent the interactions among genes, and then deploy a graph convolution encoder over it to learn low-dimensional vector representations for each gene. Next, we adopt graph pooling techniques to generate the embedding for the entire graph which encodes both the genomic fingerprints and the relevant PPI pattern of a cell line or xeno-graft. Finally, we feed the learned representation of the entire input graph to a graph decoder module to predict drug responses. We evaluate our model on three publicly available datasets: CCLE-GDSC dataset (drug responses of 459 cell lines to 45 drugs), NCI-ALMANAC dataset (drug responses of 60 cell lines to 2128 drug pairs) and NIBR-PDXE (drug responses of 93 xeno-grafts to 17 drugs) [12] and compare our model with state-of-the-art baselines such as support vector machine (SVM), random forest (RF) and deep neural networks (DNN). Experimental results show that our approach can further boost the prediction performance by incorporating PPI information and achieve promising performance compared to the baselines. The major contribution of our work is summarized as follows:

- We propose a graph neural network based approach to simulate the structure of biological tissues so as to incorporate PPI information into drug response prediction.
- We design and conduct systematic and comprehensive experiments on three publicly available datasets against state-of-the-art baselines to demonstrate the superiority of including PPI information into drug response prediction.
- We explore different graph pooling strategies for predicting drug responses and we also investigate the impact of different types of PPIs in predicting drug responses.

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3, we describe our approach in detail. Section 4 presents the empirical results and Section 5 and 6 conclude the paper.

## Related Work

### Drug Response Prediction Based on Genomic Features

Great efforts have been made to model in-depth the complex interplay among multiple omics data measured from cancer cell lines or patient tumors, which creates new opportunities for identifying personalized treatment for individual patients. Machine learning models are increasingly being applied to the genomics profiles as they provide not only integrative analysis among the high dimensional data but personalized predictions of therapy responses using multi-omics panels of response-predictive biomarkers as well. In particular, Nguyen *et al.* [4] applied random forest together with optimal dimensionality reduction techniques to predict drug responses of PDXs which are molecularly profiled. Ali *et al.* [13] and Cortés *et al.* [14] applied multiple machine learning models including support vector machine, kernel based methods, random forest, elastic net and multi-view multi-task linear regression to predict anti-cancer drug responses. With the quick advances in the deep learning community, deep neural network, a powerful representation learning model has been

widely adopted in this field. In particular, Chiu *et al.* [5] and Li *et al.* [15] combined auto-encoder with deep neural networks to predict drug responses of genomically profiled cell lines and tumors. Xia *et al.* [6] utilized deep neural encoders to combine genomic features together with drug profiles to predict cell lines' responses to drug pairs. Wang *et al.* [16] applied DNN model to predict chemically induced liver toxicity endpoints from transcriptomic responses. However, none of these methods aforementioned utilizes the PPI information which has been proved to be highly correlated with the development of certain diseases including cancers.

#### Protein-Protein Interaction Networks

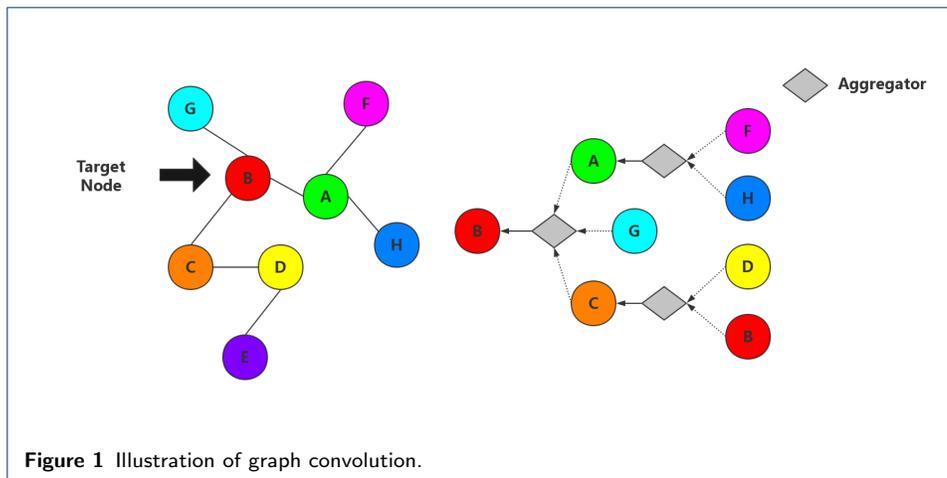
As indicated by [7], proteins within a tissue rarely act alone and tend to team up to undertake complex biological functions together through multimodal physicochemical dynamic connections. These connections which are often termed protein-protein interactions could be positive if two proteins activate each other or negative if two proteins inhibit each other [17]. Some studies [9, 10] proved that groups of genes linked by molecular interactions are more likely to have correlated gene profiles than genes chosen at random and some subnetworks of the entire PPI network could have high statistical significance for the identification of certain diseases and PPI hubs and nodes that are critical for the acquisition and maintaining characteristics of cancer essential for cell transformation have become promising therapeutic targets [8]. It is indicated that including PPI information into drug prediction could potentially boost performance.

#### Graph Neural Network and Its Applications in Bioinformatics

For the past several years, inspired by the rapid development of deep learning on grid-like inputs, learning representations on irregular graphs has enjoyed great success. In particular, the graph neural networks have been widely applied to many bioinformatic applications. Coley *et al.* [18] applied graph convolutional neural networks to predict products of organic reactions given reactants, reagents, and solvent(s). Han *et al.* [19] combined matrix factorization with graph neural networks to predict the associations between genes and diseases. Zitnik *et al.* [20] constructed a multimodal graph of proteins and drugs to model the side effects of drug combinations. Parisot *et al.* [21] represented populations as a sparse graph, where its nodes are associated with imaging-based feature vectors, while phenotypic information is integrated as edge weights and exploit graph convolutional neural network to predict the possibility of certain diseases. Song *et al.* [22] and Wee *et al.* [23] applied GNN to imaging data for classification of Alzheimer's disease. Alshahrani *et al.* [24] developed a method that uses feature learning to generate vector-based representations of phenotypes associated with an entity. Our approach is inspired by the aforementioned studies and is, as far as we know, the first one to utilize graph neural networks in drug response prediction.

## Methods

In this section, we detail our approach. We first present the rationale of graph neural network and then explain how we approach the drug response prediction problem based on it.



### Vector Representation on Graphs

[25] employ neural networks trained on large scale unlabelled data to learn high-quality vector representations which are often referred to as embeddings, of words to improve the performance of many natural language processing tasks. This notion of distributed representation of atomic objects has been transferred to many other domains, including representation learning on graphs. Our work utilizes the same mechanism as we build a graph of genes connected with edges denoting the PPIs and aim to learn quality vector representations of nodes that are aggregated to encode the entire input graph.

### Graph Convolutional Encoder

Inspired by the deep convolutional neural networks, the graph neural networks conduct convolutional operations over the neighborhood of nodes in irregular input graphs. In general, to generate new embedding of a node given the current learned embeddings in the network, GNN collects embeddings of neighboring nodes, merge them using some aggregate function and then multiply the aggregated embedding with trainable weights and this procedure could be repeated for  $K$  hops which is usually termed the depth of the graph convolution. The process is illustrated in Figure 1. In this study we use the mean variant of GraphSAGE proposed by [26] as the forward graph convolution method which are described in details in Algorithm 1. Formally, given a graph  $G = \{\mathcal{V}, \mathcal{E}\}$ , Algorithm 1 generates node embeddings as follows where  $k$  denotes the current step in the outer loop (or the depth of the graph convolution) and  $\mathbf{h}^k$  denotes the node representation at this step: First, each node  $v \in \mathcal{V}$  collects learned embeddings in the previous step (i.e.,  $k - 1$ ) of nodes in its immediate neighborhood  $\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}$  and applies aggregate function to produce a single vector  $\mathbf{h}_{\mathcal{N}(v)}^{k-1}$ . Note that  $\mathbf{h}^0$ , which is the initial node embedding is the actual input node features. The aggregated neighborhood embedding,  $\mathbf{h}_{\mathcal{N}(v)}^{k-1}$ , is then concatenated with the node's current embedding  $\mathbf{h}^{k-1}$  and this concatenated vector is fed through a fully connected layer with nonlinear activation unit  $\sigma$ , which produces the representation  $\mathbf{h}^k$  that will be used in the next step. We apply batch normalization [27] after each graph convolution layer.

**Algorithm 1:** GraphSAGE embedding generating algorithm [26]

---

**input :** Graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ ; input features  $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$ ; depth  $K$ ; weight matrices  $\{\mathbf{W}^k, \forall k \in [1, K]\}$ ; activation unit  $\sigma$ ; differentiable aggregator functions  $\{AGG_k, \forall k \in [1, K]\}$ ; neighborhood function  $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

**output:** Vector representations  $\mathbf{z}_v, \forall v \in \mathcal{V}$

- 1  $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$ ;
- 2 **for**  $k \leftarrow 1$  **to**  $K$  **do**
- 3     **for**  $v \in \mathcal{V}$  **do**
- 4          $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow AGG(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$
- 5          $\mathbf{h}_v^k \leftarrow \sigma(\{\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\})$
- 6      $\mathbf{h}_v^k \leftarrow \text{NORMALIZE}(\mathbf{h}_v^k), \forall v \in \mathcal{V}$
- 7  $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

---

**Neighborhood Subsampling**

We adopt the neighborhood sampling strategy introduced in [26] as this not only keeps the computational footprint of each batch fixed but also reduces model complexity. Specifically, instead of using information from all neighbor nodes, we sample in advance a fixed number  $N$  of nodes from the neighborhood for each node at each graph convolution layer before we perform the forward embedding generation process where  $N$  is a hyperparameter which is discussed in Table 2. If a node has less than  $N$  neighbors, we aggregate embeddings from all the nodes in its neighborhood.

**Graph Pooling Strategy**

As described in the previous section, the graph neural encoder generates embeddings that encode both the original node features and the graph topological information as well by propagating information among node neighborhoods. However, in order to correctly identify the label associated with the entire graph, we need to design how to aggregate embeddings of all the nodes in the graph to generate the final representation of the entire graph. Thus we explore two different graph pooling strategies and evaluate them on three datasets.

*Hierarchical Differentiable Graph Pooling*

Ying et al. [28] proposed an end-to-end differentiable graph pooling strategy, Diffpool, that learns hierarchical representations of graphs by defining a procedure to gradually coarsen input graphs, which is illustrated in Figure 2. Apart from defining a GNN module that is specialized in learning node embeddings, Diffpool defines a separate GNN module to learn node assignment at each pooling layer. Formally, given the adjacency matrix  $A^l \in \mathbb{R}^{n_l \times n_l}$  and the node embeddings  $X^l \in \mathbb{R}^{n_l \times d}$  at the  $l$ th pooling layer where  $n_l$  is the number of nodes in the current pooling layer and  $d$  the embedding dimension, the embedding GNN module learns the node embedding as follows:

$$Z^l = \text{GNN}_{emb}^l(A^l, X^l) \quad (1)$$

where  $Z^l$  is the intermediate node embeddings at the  $l$ th pooling layer. The pooling GNN module on the other hand takes the same input and learns an assignment

matrix that is used to cluster nodes to coarsen the graph:

$$S^l = \text{GNN}_{\text{pool}}^l(A^l, X^l) \quad (2)$$

The output dimension of each GNN pooling module defines the maximum number of clusters in this layer and is a hyperparameter of the model. We set the output dimension of the last GNN pooling module to be 1, i.e., to assign all the nodes in  $L - 1$ th layer to a single cluster in the  $L$ th layer. The assignment matrix  $S^l \in \mathbb{R}^{n_l \times n_{l+1}}$  specifies how the nodes in the  $l$ th pooling layer will be clustered into nodes in the  $l + 1$ th layer. In particular, each row of  $S^l$  corresponds to a node (cluster) in the  $l$ th pooling layer and each column of  $S^l$  a node (cluster) in the  $l + 1$ th layer. Node representations and adjacency matrix of the  $l + 1$ th layer are generated using the following two equations:

$$X^{l+1} = S^{lT} Z^l \in \mathbb{R}^{n_{l+1} \times d} \quad (3)$$

$$A^{l+1} = S^{lT} A^l S^l \in \mathbb{R}^{n_{l+1} \times n_{l+1}} \quad (4)$$

Note that matrix  $A^{l+1}$  is a real matrix that represents a fully connected edge-weighted graph and  $A_{ij}^{l+1}$  denotes the connectivity strength between cluster  $i$  and cluster  $j$ .  $A^{l+1}$  and  $X^{l+1}$  can be taken as the input of the next pooling layer and by iterating over Equation 3 and 4, the original input graph is coarsened until the final representation of the graph is produced which is fed through a fully connected neural network for classification or regression. To guide the pooling process, we add the link prediction objective which encourages nearby nodes to be pooled together. In particular, we define layer loss as follows:

$$L_{LP} = \|A^l, S^l S^{lT}\|_F \quad (5)$$

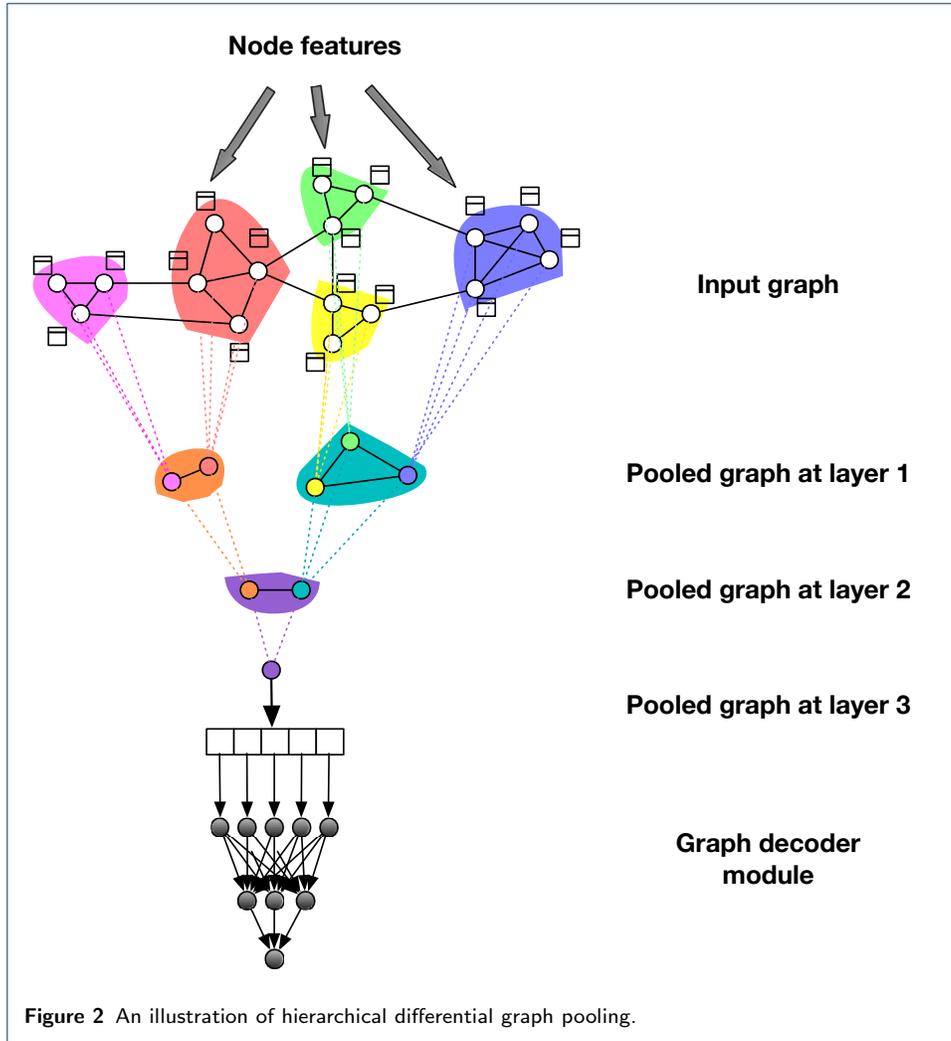
where  $\|\cdot\|$  denotes Frobenius norm. In addition, we want the node assignment to be clearly defined in each pooling layer, i.e., we want each row in the assignment matrix  $S$  to be close to a one-hot vector. So we add the entropy loss introduced in [28] defined as follows:

$$L_E = \frac{1}{n} \sum_{i=1}^N H(S_i) \quad (6)$$

where  $H(\cdot)$  denotes the entropy function and  $S_i$  is the  $i$ th row of  $S$ .

#### *Self-Attention Graph Pooling*

The second pooling strategy we evaluate is based on the attention mechanism [29, 30, 31] that has recently achieved great success in many deep learning applications. Specifically, an attention function maps a query and a set of key-value pairs to an output which is a weighted sum of the values and the weight assigned to each value

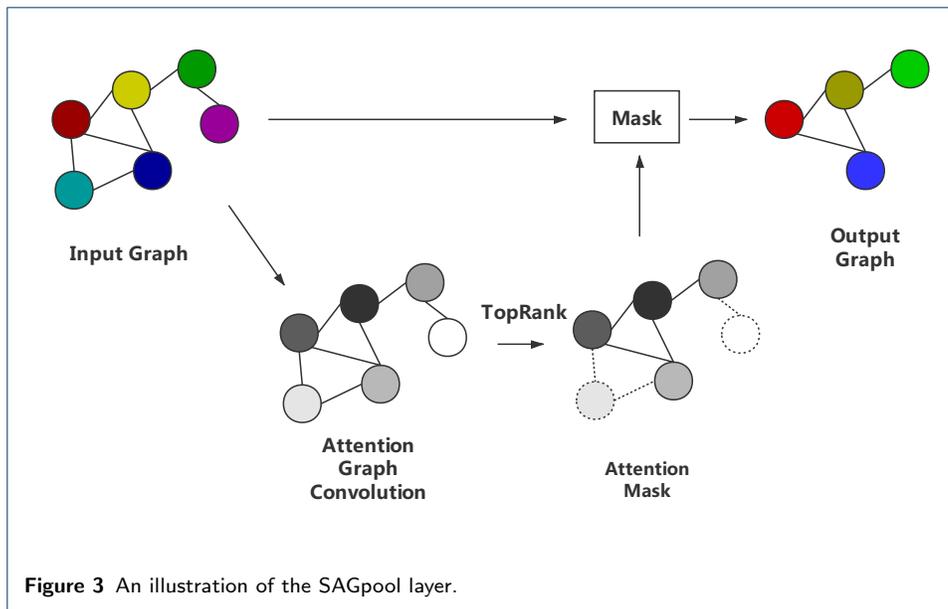


is determined by a compatibility function of the query with the corresponding key. Furthermore, Lee et al. [32] proposed self-attention based graph pooling, SAGpool, which defines a GNN module to learn attention scores for all the nodes using both node features and topological features and the learned attention scores are used to only choose a subset of the original node sets that have the maximum attention scores during pooling. The pooling layer is illustrated in Figure 3. Formally, given the adjacency matrix  $A \in \mathbb{R}^{N \times N}$  and node embeddings  $X \in \mathbb{R}^{N \times d}$ , the attention GNN module learns an attention score for each node as follows:

$$Z = \sigma(\text{GNN}(X, A)) \in \mathbb{R}^{N \times 1} \quad (7)$$

where  $N$  is the number of nodes in the input graph,  $d$  is the node feature dimension and  $\sigma$  is the nonlinear activation function. Given the attention scores of each node, SAGpool selects the top  $\lceil kN \rceil$  nodes with the highest  $Z$  scores:

$$\text{idx} = \text{topRank}(Z, \lceil kN \rceil), Z_{\text{mask}} = Z_{\text{idx}} \quad (8)$$



where pooling ratio  $k$  is a hyperparameter,  $\text{topRank}()$  is the operation to return the indices of the top  $\lceil kN \rceil$  values,  $\cdot \text{idx}$  is the indexing operation and  $Z_{\text{mask}}$  is the feature attention mask. The pooled graph is obtained as follows:

$$X^* = X[\text{idx}, :], X_{\text{out}} = X^* \odot Z_{\text{mask}}, A_{\text{out}} = A[\text{idx}, \text{idx}] \quad (9)$$

where  $X[\text{idx}, :]$  is the row-wise indexed feature matrix,  $\odot$  is the broadcasted element-wise product, and  $A[\text{idx}, \text{idx}]$  is the row-wise and column-wise indexed adjacency matrix.  $X_{\text{out}}$  and  $A_{\text{out}}$  are the new feature matrix and the corresponding adjacency matrix.

We also adopt the ReadOut layer [33] after graph pooling to generate a fixed representation which is then fed to a fully connected neural network for downstream tasks.

### Graph Decoder Module

After obtaining the graph representation through graph pooling, we feed it to a graph decoder module which contains a fully connected neural network with non-linear activation function between each layer. In particular, for the NIBR-PDXE dataset, we use an architecture of two (512, 128 neurons for each layer) or three layers (1024, 256, 128 neurons for each layer) and for the CCLE-GDSC and NCI-ALMANAC, we use a four layer architecture with 2048, 1024, 512, 128 neurons in each layer.

## Results

In this section, we describe how we design and conduct the experiments to empirically evaluate our model. We first describe the datasets, then the baseline models and training details, and finally the performance comparison and model interpretation.

## Datasets

We use three publicly available datasets which contain drug responses of PDXs and cell lines. We describe them in details in the following sections. As mentioned in many previous studies [14, 15, 6], including drug descriptor features will overshadow the impact of genomic features, in this study, we focus on the impact of genomic features and only use one-hot encoding features for drugs and do not utilize any drug feature extractors. For virtual drug screening, our approach could be easily extended to include more features. For example, we can add drug encoders to generate drug embeddings which could be combined with genomic embeddings before we feed them to the final decoder module.

### *NIBR-PDXE Dataset*

The NIBR-PDXE dataset contains the profile information as well as recorded treatment responses of about 400 PDX samples each of which is profiled at three omic levels: gene expression (GEX), copy number and detected somatic mutations. The raw treatment responses include the percentage of tumor volume change relative to tumor volume at the beginning of treatment of each treated PDX recorded every 3-4 days. We preprocess the raw data following the procedures in [4, 12] to generate four different molecular profiles for each PDX:

- **Gene Expression and Copy Number.** For these two profiles, we directly use the recorded gene expression values of 22665 genes and copy number values of 23853 genes from the raw data.
- **Detected Somatic Mutation.** For each gene considered, we create a feature value of 1 if a somatic mutation is reported in the raw data and 0 for wild-type, resulting in a binary profile of 15232 features.
- **Copy Number Alteration.** The NIBR-PDXE raw data also includes a categorical profiling of copy numbers. In particular, we binarized copy number data so that the Copy-Number Alteration (CNA) feature of a gene has a value of 1 for aberrant copy number (Amp8, Amp5 or Del0.8) and 0 otherwise.

Following the procedures in [12], we also preprocess the raw treatment response data based on recorded Best Response (BR, the minimum value of volume change for  $t \geq 10$  days) and Best Average Response (BAR, the minimum value of the set of average responses spanned by all  $t$  values with  $t \geq 10$  days) values. Specifically, if its  $BR < -95\%$  and  $BAR < -40\%$ , one PDX was categorized as Complete Response (CR); Partial Response (PR) if  $-95\% \leq BR < -50\%$  and  $-40\% \leq BAR < -20\%$ ; Stable Disease (SD) if  $-50\% \leq BR < 35\%$  and  $-20\% \leq BAR < 30\%$ ; otherwise, it was categorized as Progressive Disease (PD). As does in [12], we further group them into two categories. That is, CR, PR and SD are considered as SENSITIVE while PD is considered as RESISTANT. Since only a part of all the PDX samples are both molecularly and treatment-response profiled, we only consider a subset of the whole data that has sufficient data points (more than 35 except for one cancer treatment pair) as does in [4]. Specifically, we use a set of 13 treatments administered to breast cancer (BRCA) PDX samples and another set of 13 treatments to colorectal cancer (CRC) PDX samples. Detailed sensitive and resistant sample ratios for each cancer-treatment pair are recorded in table 1.

**Table 1** Response rate for each cancer-treatment pair. BRCA represents breast cancer and CRC colorectal cancer.

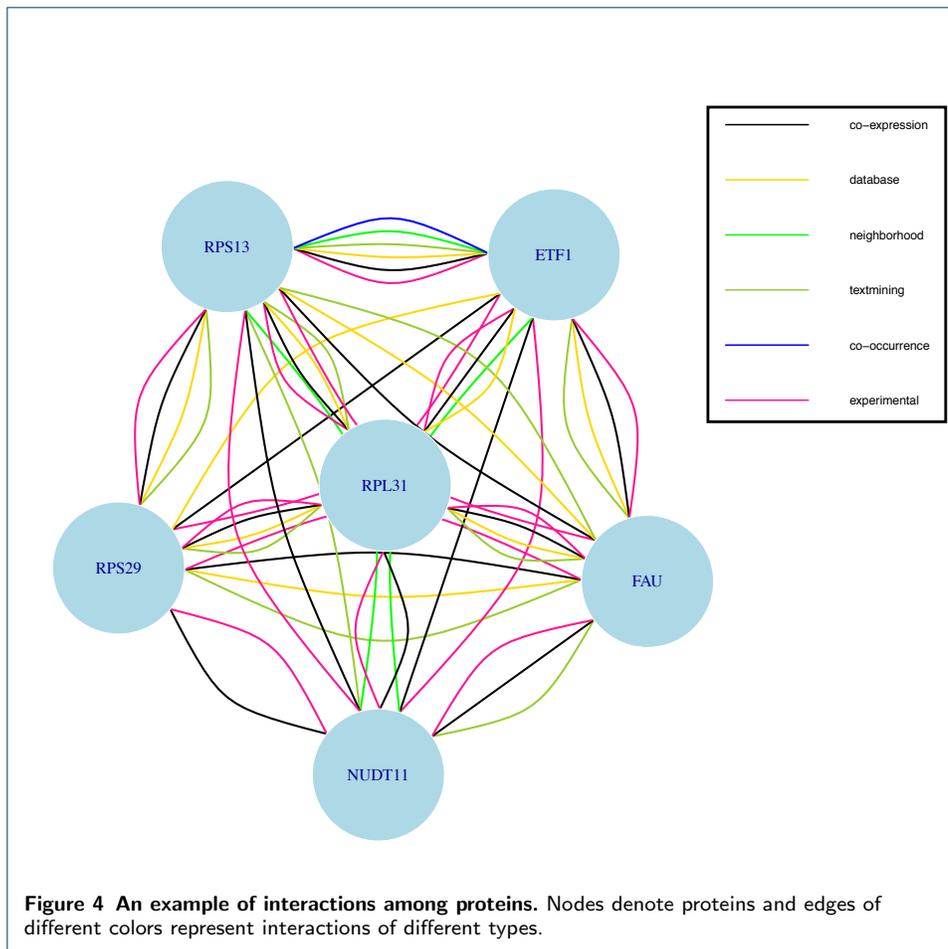
Cancer	Treatments	Sensitive	Total	Response Rate
BRCA	BGJ398	14	38	0.37
BRCA	BKM120	30	38	0.79
BRCA	BYL719	24	38	0.63
BRCA	BYL719 + LJM716	33	38	0.87
BRCA	CGM097	12	38	0.32
BRCA	CLR457	27	38	0.71
BRCA	HDM201	14	38	0.37
BRCA	INC424	14	37	0.38
BRCA	LEE011	23	38	0.61
BRCA	LKA136	12	38	0.32
BRCA	LLM871	23	38	0.61
BRCA	binimetinib	22	38	0.58
BRCA	paclitaxel	26	38	0.68
CRC	BKM120	26	40	0.65
CRC	BYL719	26	42	0.62
CRC	BYL719 + LJM716	28	40	0.70
CRC	CGM097	12	37	0.32
CRC	CLR457	29	41	0.71
CRC	HDM201	13	40	0.33
CRC	LEE011	25	42	0.60
CRC	LKA136	15	40	0.38
CRC	binimetinib	28	42	0.67
CRC	cetuximab	16	41	0.39
CRC	CKX620	29	39	0.74
CRC	encorafenib	12	41	0.29
CRC	LFW527 + binimetinib	16	31	0.52

#### *CCLE-GDSC Dataset*

The CCLE [1] project recorded the genomic profiles of 1457 human cell lines and the GDSC project drug sensitivity data of 1000 human cell lines measured by the half maximal inhibitory concentration ( $IC_{50}$ ). The  $IC_{50}$  is measured in  $\mu M$  and has been transformed using natural log function. A low  $IC_{50}$  value indicates that the given cell line is sensitive to the given drug while a high  $IC_{50}$  value means the opposite. Since there is a mismatch (not all the cell lines have both genomic profiles and drug response data) between the two projects, we only select a subset of all the cell lines that have both genomic features and drug response data. Furthermore, we only select the cell lines that have enough drug response data as we do for the NIBR-PDXE dataset and this results in a dataset of 21,465 cell line - treatment entries (477 cell lines  $\times$  45 drugs). As regards to the genomic features, we adopt the same criteria used in [5] to remove genes with low information; we remove a gene if its mean  $< 1.0$  or st.dev.  $< 0.5$ , resulting in a feature vector of 14,892 genes. We also include the gene mutation information following procedures in [5] and this results in a feature vector of 17,960 features. We combine gene expression with mutation information as node features for GNDR and we use 0 for the mismatched genes between the two types of features.

#### *NCI-ALMANAC Dataset*

The third dataset we use is from the NCI-ALMANAC [34] project which records the drug pair responses of 60 cell lines to 105 drugs. The response value was relative tumor growth described in the standard NCI-60 testing protocol [6]. We followed the procedures in [6] to convert the growth inhibition percentage to fraction, ranged from -1 to slightly above 1, with -1 representing 100% lethality, 0 representing total



inhibition, and 1 representing unabated growth compared to control tumor tissue. Since many drug pairs are not tested on all cell lines, we only select the drug pairs that are tested on all 60 cell lines, resulting in a dataset of 127,680 entries (60 cell lines  $\times$  2128 drug pairs). We only predict the best growth inhibition for a given cell line and treatment pair, i.e., we do not include the drug concentration as a separate input feature as does in [6]. As for the genomic features of the cell lines, we use the gene transcript expression levels averaged from five microarray platforms from the NCI's CellMiner [35]. Each cell line is characterized by 25,722 gene features. We also filled the missing values with the mean over cell lines and then used min-max scaling to normalize each feature to the  $[0, 1]$  range as suggested in [6].

#### *Protein-Protein Interaction Data*

We download the PPI data from STRING [36] which includes both direct (physical) and indirect (functional) associations; they come from various sources including computational prediction, knowledge transfer between organisms, and interactions aggregated from other databases. An example of PPIs is illustrated in Figure 4. We consider in total 8 different types of interactions: 'neighborhood', 'gene fusion', 'co-occurrence', 'co-expression', 'experimental', 'database', 'textmining', and 'combined' (computed based on all the 7 previous types of interactions following

procedures in [37]). We analyze the impact of different types of interactions in later sections and we use the ‘combined’ scores for all the experiments unless otherwise specified. We use the interaction scores as edge weights in graph convolution and we describe the sources of different types of interaction in later sections.

### Baselines

In this study, we compare our model with three state-of-the-art baseline models which are support vector machine, random forest and deep neural networks.

#### *Support Vector Machine*

The first baseline model we use is the support vector machine (SVM) which seeks to minimize the hinge loss defined as follows:

$$h(\mathbf{x}_i, y_i | \mathbf{w}, b) = \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i - b)) \quad (10)$$

where  $\mathbf{x}$  is the input feature and  $\mathbf{w}$  and  $b$  the model weights. We employ the gaussian radial basis kernel function and add  $L2$  regularizer to avoid overfitting.

#### *Random Forest*

Random forest, an ensemble model of decision trees with random feature sub-sampling, has been widely applied to many bioinformatic applications. For the NIBR-PDXE dataset, we combine the optimal model complexity approach which uses two-sided Fisher’s exact test and t-test to rank features and select the top  $k$  features, which is consistent with [4]. For the other two datasets, we include all the features as we are solving a regression problem and cannot use the two tests aforementioned.

#### *Deep Neural Networks*

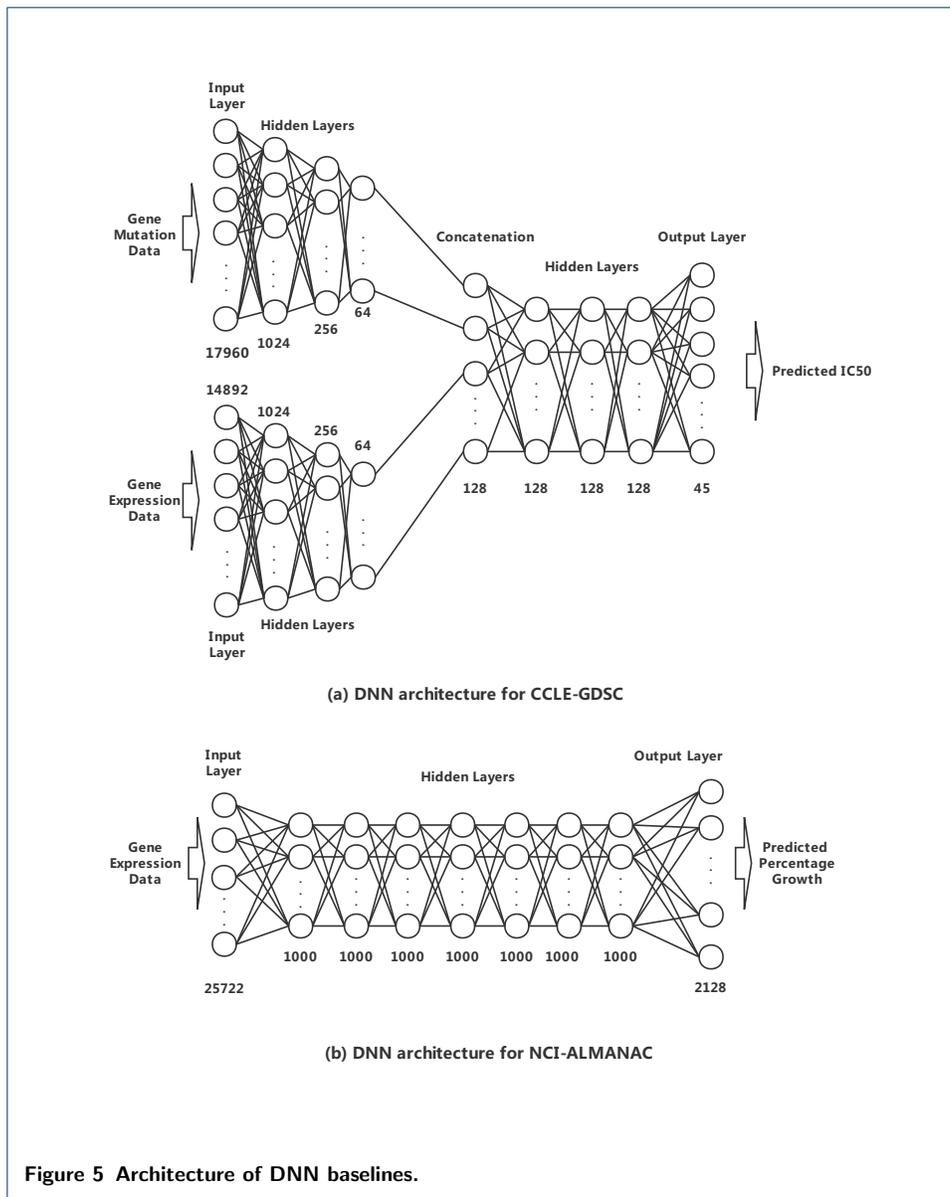
We adopt two architectures from [5] and [6] as they are also evaluated on the CCLE-GDSC and NCI-ALMANAC datasets respectively. Note that for the second architecture, we did not include the drug encoder as we want to focus on the impact of the genomic features. The two architectures are illustrated in Figure 5.

### Metrics of Performance

In this section, we describe the metrics that we use to compare the performance of our model with the baseline models on three datasets. For NIBR-PDXE dataset, we employ the Matthews Correlation Coefficient(MCC) used in our baseline model [4]:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN) * (FN + TN) * (TN + FP) * (FP + TP)}} \quad (11)$$

where TP, TF, FP, FN stand for true positive, true negative, false positive, false negative respectively. For the other two datasets which involve a regression problem,



we use the mean squared error (MSE) and mean absolute error (MAE) to compare the model performance. They are defined as follows:

$$MSE = \frac{1}{N} \sum_i^N (y_{pred}^i - y_{true}^i)^2 \tag{12}$$

$$MAE = \frac{1}{N} \sum_i^N |y_{pred}^i - y_{true}^i| \tag{13}$$

where  $N$  is the number of data points,  $y_{pred}$  is the model prediction and  $y_{true}$  the true response. We do not use  $R^2$  in our study as indicated in [15] that when the data range considered is small,  $R^2$  cannot reflect the goodness of the prediction performance of the model, and it may get worse even the model accuracy is actually improved.

### Training Details

In this section, we describe details as regards to how to train our model. As mentioned in [38], different splits of data can affect GNN model performance. For the NIBR-PDXE dataset, we use nested LOOCV to make a fair comparison with the baseline model and for the other two datasets, we perform 5-fold cross validation. On the CCLE-GDSC and NCI-ALMANAC datasets, we further perform leave-one-tissue-out (LOTO) validation method [14] to test our model’s capability on unseen types of tissues. In other words, cell lines from the same tissue of origin are left from the training data and set as the test data. Note that since we do not include any drug descriptor features, we do not perform leave-one-compound-out (LOCO) validation[14]. We repeated our experiments on each dataset for 10 random runs and reported the mean performance with standard deviation. We use the Adam optimizer with weight decay for model optimization. Besides, we adopt cross entropy loss function for classification and mean squared loss function for regression. For the NIBR-PDXE dataset, we use nested LOOCV to determine the number of epochs for each cancer-treatment pair (the best values were found via grid search). For the other two datasets, we run our model for 2000 epochs with early stopping applied when the validation accuracy starts to drop and we use a patience value of 30. To minimize overfitting, we further employ  $L2$  regularization and adopt dropout procedures. The optimal parameters are obtained by grid search and we list ranges of the major hyperparameters in Table 2. We use Relu [39] as the non-linear activation function for DIFFpool and tanh for SAGpool.

**Table 2** The grid search space for the hyperparameters.

Hyperparameter	Range
Learning rate	$10^{-1}$ , $10^{-2}$ , $10^{-3}$ , $10^{-4}$ , $10^{-5}$
Decaying rate	0.95, 0.99
Node embedding dimension	20, 50, 100, 200, 500
Number of layers of graph convolution	2, 3, 5, 7
Number of sampled neighbor nodes	15, 50, 100, 200, 300, 500
Dropout rate	0.1, 0.3, 0.5
Number of pooling layers	1, 2, 3
Pooling ratio	0.0025, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5

### Performance Comparison with Other Models

In this section, we present the empirical evaluation of our model against the baseline models using the metrics defined in the previous section on three datasets.

#### *Predicting Drug Responses for Breast Cancer and Colorectal Cancer on NIBR-PDXE*

We first compare our model with baselines models on the NIBR-PDXE dataset. In order to make a fair comparison with the random forest baseline [4], we train a separate model for each cancer-treatment pair and adopt the same feature ranking

strategy as does in [4] and only take the top  $k$  nodes as the model input for our GNNDR. In particular, for the baseline random forest model, we use the exact hyperparameters specified in the baseline and only use the molecular profile that reportedly produces the best performance in the baseline for each cancer-treatment pair which is also taken as input for our model. And for  $k$ , we find out that a value of 25 would work well for most of the cancer-treatment pairs. Since no deep neural networks have been explored on this dataset before, we use the graph decoder module of GNNDR as the baseline neural network model. Table 3 compares GNNDR with the baseline models on two types of cancers and numbers reported are averaged over 10 runs. We can see from Table 3 that GNNDR significantly outperforms the

**Table 3** MCC on NIBR-PDXE dataset. BRCA represents breast cancer and CRC colorectal cancer.

Cancer	Treatments	SVM	NN	RF	GNNDR-DIFF	GNNDR-SAG
BRCA	BGJ398	0.06±0.08	0.12±0.09	0.12±0.06	0.16±0.09	<b>0.21±0.03</b>
BRCA	BKM120	0.02±0.07	0.02±0.10	0.10±0.09	<b>0.13±0.08</b>	0.12±0.04
BRCA	BYL719	0.02±0.06	0.04±0.08	0.35±0.05	<b>0.47±0.03</b>	0.46±0.08
BRCA	BYL719+LJM716	0.01±0.06	<b>0.03±0.05</b>	-0.05±0.06	0.01±0.04	-0.05±0.04
BRCA	CGM097	0.17±0.08	0.03±0.05	<b>0.26±0.06</b>	0.14±0.09	0.18±0.11
BRCA	CLR457	-0.21±0.05	-0.12±0.04	-0.24±0.04	0.05±0.06	<b>0.11±0.03</b>
BRCA	HDM201	0.11±0.11	0.03±0.11	0.17±0.09	0.24±0.10	<b>0.25±0.05</b>
BRCA	INC424	0.17±0.09	0.21±0.09	0.21±0.08	0.26±0.09	<b>0.30±0.07</b>
BRCA	LEE011	0.03±0.07	-0.04±0.08	<b>0.31±0.06</b>	0.19±0.04	0.25±0.03
BRCA	LKA136	0.43±0.08	0.47±0.07	0.48±0.05	0.51±0.07	<b>0.52±0.10</b>
BRCA	LLM871	-0.07±0.06	0.02±0.05	0.03±0.04	0.01±0.05	<b>0.09±0.06</b>
BRCA	binimetinib	0.49±0.05	0.62±0.04	0.62±0.05	0.62±0.04	<b>0.64±0.05</b>
BRCA	paclitaxel	0.37±0.06	0.22±0.04	0.48±0.07	0.49±0.06	<b>0.53±0.11</b>
CRC	BKM120	0.01±0.09	0.08±0.07	-0.10±0.08	0.03±0.09	<b>0.09±0.06</b>
CRC	BYL719	0.02±0.07	0.02±0.06	0.06±0.03	0.09±0.06	<b>0.12±0.07</b>
CRC	BYL719+LJM716	0.11±0.02	0.15±0.06	0.05±0.04	0.14±0.06	<b>0.16±0.04</b>
CRC	CGM097	0.17±0.06	0.21±0.05	0.21±0.06	0.23±0.08	<b>0.29±0.08</b>
CRC	CLR457	0.06±0.12	0.00±0.06	0.13±0.08	<b>0.17±0.12</b>	0.14±0.10
CRC	HDM201	-0.07±0.08	0.05±0.07	0.01±0.02	0.03±0.02	<b>0.09±0.04</b>
CRC	LEE011	-0.13±0.07	-0.08±0.02	<b>0.27±0.03</b>	0.12±0.05	0.16±0.05
CRC	LKA136	-0.09±0.05	0.01±0.05	0.18±0.02	0.11±0.02	<b>0.20±0.04</b>
CRC	binimetinib	0.01±0.06	0.01±0.04	0.02±0.04	0.06±0.06	<b>0.08±0.04</b>
CRC	cetuximab	0.37±0.05	0.38±0.04	0.45±0.04	<b>0.51±0.05</b>	0.49±0.07
CRC	CKX620	0.03±0.06	0.03±0.04	<b>0.05±0.05</b>	0.03±0.05	-0.01±0.03
CRC	encorafenib	-0.05±0.07	0.02±0.09	-0.12±0.08	<b>0.05±0.12</b>	-0.07±0.14
CRC	LFW527+binimetinib	0.09±0.05	0.27±0.07	0.14±0.09	<b>0.30±0.12</b>	0.17±0.09

two baselines in most of the cancer-treatment pairs (21 out of 26). Particularly, on four cancer-treatment pairs, namely, BRCA-LKA136, BRCA-binimetinib, BRCA-paclitaxel, and CRC-cetuximab, GNNDR achieves mcc values greater than 0.5. Besides we observe on this dataset that GNNDR-SAG seems to perform better than GNNDR-DIFF, which could be explained that since the number of input genes are limited by correlation ranking, the corresponding nodes in the input graph lack topological similarities to learn useful gene clustering patterns whereas nodes display significantly different degrees of importance that is particularly suitable for SAGpool. We further discuss this problem in the model interpretation section.

#### *Prediction of Growth Inhibition on CCLE-GDSC and NCI-ALMANAC*

Different from the NIBR-PDXE data, prediction of drug responses on the CCLE-GDSC and NCI-ALMANAC datasets are essentially a regression problem, thus we cannot use the feature ranking techniques in [4]. Instead, we include all the features into the input. We use 5-fold cross validation and report the average performance over 10 runs. Results are described in Table 4 and Table 5.

**Table 4 Prediction Performance on CCLE-GDSC.**

Algorithm	5-fold CV		LOTO	
	MSE	MAE	MSE	MAE
SVM	2.45 ± 0.05	1.20 ± 0.03	2.51 ± 0.0	1.24 ± 0.0
DNN	1.89 ± 0.05	1.01 ± 0.03	1.98 ± 0.02	1.03 ± 0.03
RF	1.53 ± 0.03	0.94 ± 0.02	<b>1.61 ± 0.02</b>	<b>0.94 ± 0.02</b>
GNNDR-DIFF	<b>1.46 ± 0.02</b>	<b>0.92 ± 0.01</b>	1.64 ± 0.03	0.95 ± 0.02
GNNDR-SAG	1.48 ± 0.03	0.93 ± 0.02	1.67 ± 0.02	0.97 ± 0.02

**Table 5 Prediction Performance on NCI-ALMANAC.**

Algorithm	5-fold CV		LOTO	
	MSE	MAE	MSE	MAE
SVM	0.30 ± 0.04	0.38 ± 0.03	0.34 ± 0.0	0.40 ± 0.0
DNN	0.22 ± 0.02	0.28 ± 0.04	0.26 ± 0.02	0.36 ± 0.02
RF	0.19 ± 0.03	0.26 ± 0.05	0.18 ± 0.02	0.26 ± 0.03
GNNDR-DIFF	<b>0.11 ± 0.02</b>	<b>0.20 ± 0.03</b>	<b>0.14 ± 0.03</b>	<b>0.23 ± 0.02</b>
GNNDR-SAG	0.14 ± 0.02	0.23 ± 0.03	0.16 ± 0.03	0.24 ± 0.02

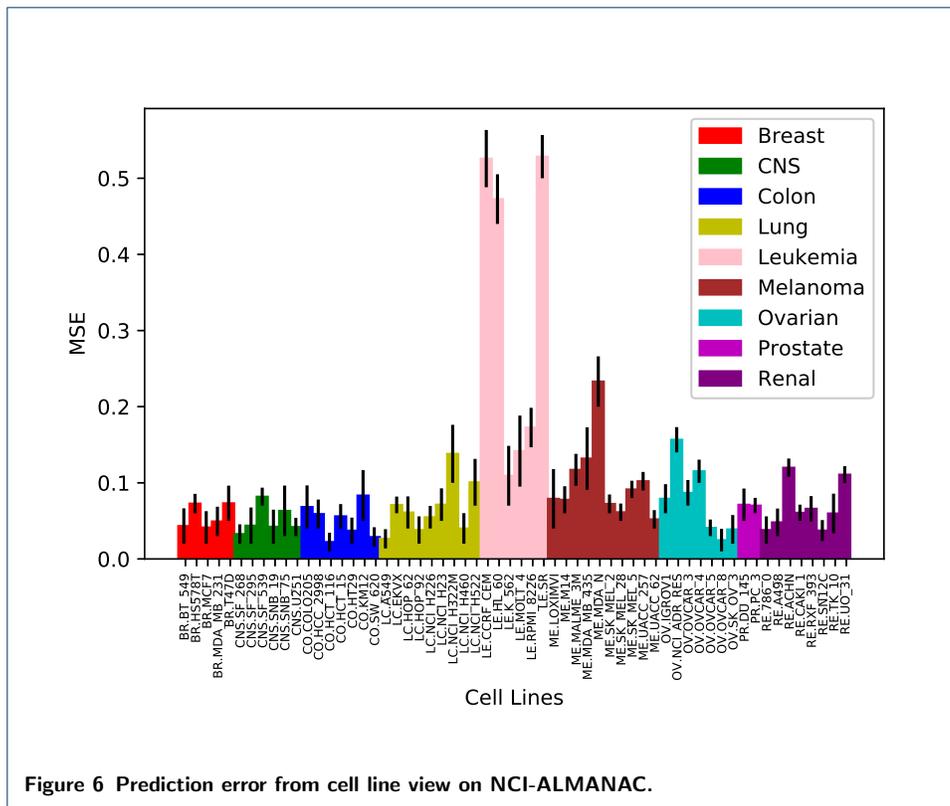
From the two tables we can see that our model significantly outperforms SVM and DNN on both datasets and achieves comparable results as random forest. We also notice that GNNDR-DIFF performs better on the two regression datasets than GNNDR-SAG. A possible explanation for this could be that for the two regression datasets, the numbers of nodes of input graphs are much larger than that of NIBR-PDXE and many of the nodes have similar embeddings induced by densely connected subgraphs and are likely to be grouped into clusters to generate a coarsened graph, which is more suitable for the DIFFpool strategy whereas SAGpool tends to be distracted by too many nodes and may not be able to learn meaningful attention weights over a large number of input nodes.

In addition to performance comparison with the baseline models, we carry out an error analysis from the perspective of cell lines and drugs respectively. Specifically, on the NCI-ALMANAC dataset, we calculate the prediction error over all the drug pairs for each cell line and plot the results in Figure 6 and on the CCLE-GDSC dataset, we perform the error analysis from the drug view and plot the averaged MSE over all the cell lines for each drug in Figure 7. From Figure 6 we can see that the prediction errors for each cancer type are generally small except for one type which is Luekemia. In the Figure 7, however, the model displays significant performance differences among drugs.

## Utilization of Protein-Protein Interaction Information

### *Effectiveness of Protein-Protein Interaction Information*

We conduct two sets of experiments to demonstrate that our model can integrate the PPI information to improve prediction performance. In the first set of experiments, we compare our model against one that has no graph neural encoder, i.e., we directly feed the input features to the graph decoder module. And in the second set of experiments, instead of using the PPI network, we change the adjacency matrix to an identity matrix, i.e., we do not aggregate information from neighborhood for each node. In both cases, PPI information is not utilized. We compare on the CCLE-GDSC and NCI-ALMANAC datasets and results are reported in Figure 8. From the figure we can see that our model that includes the PPI information significantly outperforms the two variants that do not utilize PPI information, which proves the effectiveness of integrating PPI information into drug response prediction.



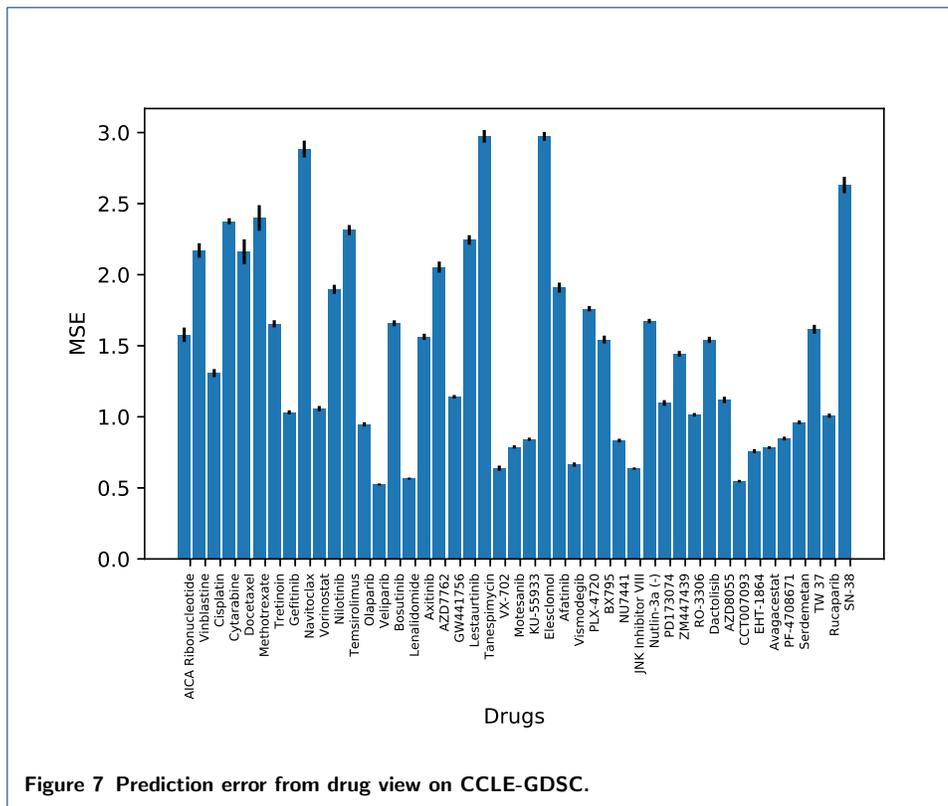
*Types of Protein-Protein Interaction*

As we mentioned earlier, there are multiple types of PPIs obtained from various sources which could form different PPI graphs and play different roles in drug response prediction. We carry out a study on the CCLE-GDSC dataset to further investigate this. First, the topological information of different types of interaction networks of the CCLE-GDSC dataset is listed in Table 6. There are in total 8 types of PPIs considered in this study where ‘database’ and ‘experimental’ represent known interactions from either curated databases or experiments (copurification, co-crystallization, Yeast2Hybrid, Genetic Interactions, etc.), ‘fusion’, ‘neighborhood’ and ‘co-occurrence’ denote interactions predicted by computational methods, and ‘textmining’ and ‘co-expression’ stand for interactions obtained from unsupervised text mining and gene co-expression network [36]. The ‘combined’ scores are computed using scores of the aforementioned 7 types following the procedures described in [37]. We evaluate the impact of different types of interactions

**Table 6** Node degrees of different types of PPI graphs on the CCLE-GDSC dataset.

Interaction Type	degree mean	degree std
combined	339	456
neighborhood	23	69
fusion	1.14	0.82
co-occurrence	2.38	7.37
co-expression	240	367
experimental	173	343
database	23	46
textmining	289	426

by running two GNNDR variants (SAGpool and DIFFpool) on the CCLE-GDSC



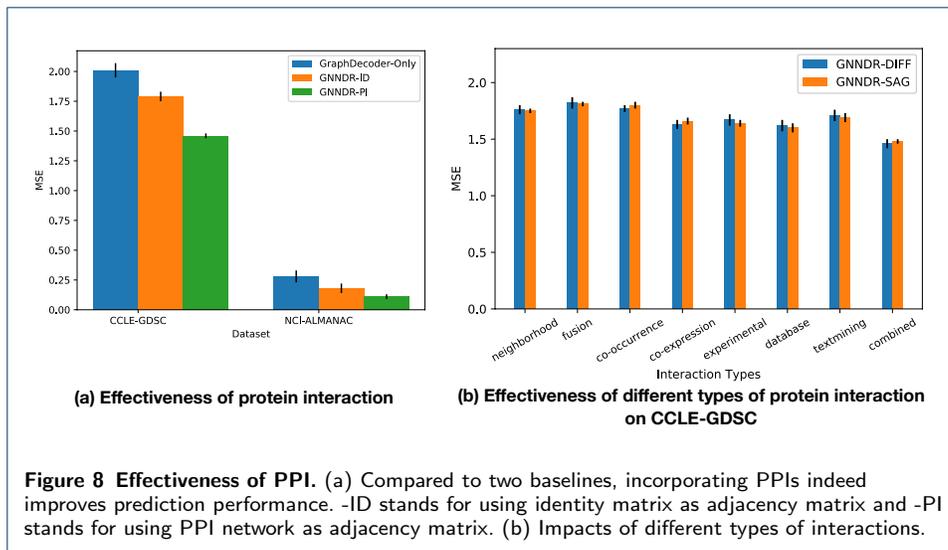
dataset. For both variants, we use the parameters we find best for the ‘combined’ score. The MSEs of different interactions are shown in Figure 8. We can see that the ‘combined’ score produces the best performance whereas ‘neighborhood’, ‘fusion’, and ‘co-occurrence’ produce the worst. Besides, it seems that the performance achieved by using the interaction information obtained from text mining is not as good as that of using experimentally determined interactions or interactions from curated databases.

### Model Interpretation

Many deep learning models achieves great performance improvement with little rationale behind the black box functions well explained. Our approach, however, with the graph pooling mechanism, offers better interpretability. In the following section, we analyze the graph pooling procedures so as to shed some light upon the improved model performance.

### Gene Clustering by DIFFpool

We first investigate the gene clustering performed by DIFFpool. By manually checking some of the node assignment on all the three datasets, we observe that DIFFpool tends to collapse densely connected subgraphs into clusters. This implies that some groups of genes that tend to function together via dynamic connections are closely related to each other in terms of interactions (For example, they form dense, clique-like subgraphs), which facilitates message passing among them, and are more likely to have similar embeddings. This is consistent with the fact that some genes are

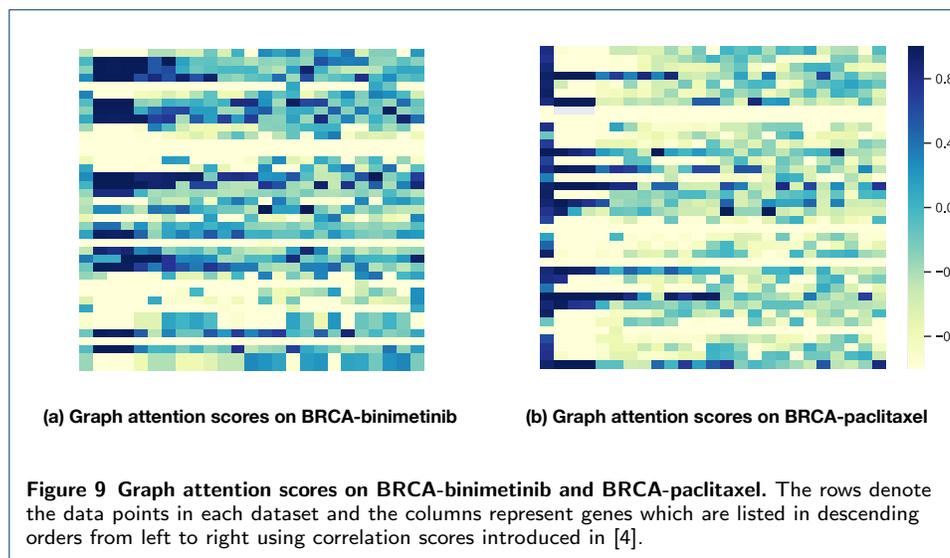


akin to each other in terms of structural similarity and likely undertake complex biological functions collectively. Besides, as indicated in [8, 10], hubs or subnetworks in the PPI networks could be highly related to the development of cancers. Therefore collapsing nodes within a densely connected subgraph into a single cluster can preserve most of the structural information while reducing noise from too many nodes. Sparse subgraphs on the other hand, are usually preserved in that the nodes are less similar due to the high diameters induced by sparsity. For example, most of the isolated nodes are pooled into single clusters since they have no communications with other nodes in the input graphs. In short, the differential graph pooling enables the model to fuse and amplify the signals carried by similar genes while maximally preserve unique sub-patterns, which may explain the performance improvement.

#### *Gene Importance by SAGpool*

The self-attention mechanism provides an alternative viewpoint of interpreting the representations learned by GNDR. Instead of utilizing the structural similarities of genes, SAGpool focuses on identifying the most important genes given genomic features as well as topological features. This could be particularly suitable when the input graphs contain no apparent densely connected subgraphs or genes in the input graphs have significantly different importance. In Figure 9, we plot heatmaps using graph attention scores on two cancer-treatment pairs that have the highest prediction performance on the NIBR-PDXE dataset (BRCA-binimetinib and BRCA-paclitaxel). Note that the attentions scores are calculated during validation and genes (columns) from left to right are listed in the order obtained by correlation ranking in [4], i.e., the leftmost column is the most correlated gene. We can see from the figure that SAGpool manages to capture most of the genes that have the highest correlation with the true labels. Furthermore, since the attention scores calculated by SAGpool is not only based on genomic features but also the topological features (the PPIs), the graph attention scores seem to have slightly different focuses compared to the correlation scores calculated using two-sided Fisher test or t-test. For example, the two most correlated gene (first two columns) in the BRCA-binimetinib

pair seem to have much lower attention scores compared to the third through 6th genes.



## Discussion

Much effort has been placed on the development of the identification of effective anti-cancer drugs using machine learning models. With the availability of more large-scale data, deep neural networks have exerted great influence on the study of drug response prediction. However, all the previous methods neglect the fact that proteins tend to perform complex biological functions together rather than act alone and there exists a huge amount of different types of interactions among proteins which have been proved to be highly related to the development of cancer and could be used as new therapeutic targets.

Inspired by advances in deep learning on grid-like input, graph neural network has been proposed to mine deep patterns from irregular graphs. Even though graph neural networks have been adopted in many bioinformatics tasks, it has not been applied to predict drug responses before. In this study, we utilize graph neural networks to incorporate PPI information into drug response prediction on three publicly available datasets for two tasks (classification and regression) and investigate the impact of different graph pooling strategies as well as different types of interactions. In addition, we discuss the roles of gene similarity as well as gene importance in drug response prediction by analyzing graph pooling procedures so as to shed some light upon the improved model performance.

## Conclusions

In this study, we propose a novel graph neural network based approach, GNNDR, to incorporate PPI information into drug response prediction. In particular, our approach exploits the fact that proteins often team up to undertake complex biological functions through intricate dynamic connections and nodes and hubs in the PPI network crucial to the development of cancer cells could be useful for the identification of anti-cancer drugs. This extension of deep learning onto irregular PPI

graphs not only possess great learning capacity but can also integrate PPI information. We design and conduct systematic and comprehensive experiments on three publicly available datasets that include both genomic profiles and drug responses of cell lines and PDXs to compare our model with state-of-the-art baselines. We also investigate the impact of different graph pooling strategies as well as different types of PPIs obtained from various sources. Empirical evaluation demonstrates the effectiveness of adding PPI information into drug response prediction and the graph pooling procedures provide some model interpretability in terms of gene similarity and gene importance. With PPI information incorporated, our model significantly outperforms the SVM and DNN baselines and achieves comparable results with the RF model. This provides a promising perspective for the clinical and medical studies on cancer treatment in the future.

## List of Abbreviations

<b>NN</b>	Neural network
<b>DNN</b>	Deep neural network
<b>GNN</b>	Graph neural network
<b>PPI</b>	Protein-protein interaction
<b>DIFFpool</b>	Hierarchical differential graph pooling
<b>SAGpool</b>	Self-attention graph pooling
<b>CCLE</b>	Cancer cell Line encyclopedia
<b>GDSC</b>	Genomics of drug sensitivity in cancer
<b>NIBR-PDXE</b>	The Novartis Institutes for BioMedical Research PDX encyclopedia
<b>PDX</b>	Patient derived xenografts
<b>GEX</b>	Gene expression
<b>CNA</b>	Copy number alteration
<b>BR</b>	Best Response, the minimum value of volume change for $t \geq 10$ days.
<b>BAR</b>	Best Average Response, the minimum value of the set of average responses spanned by all $t$ values with $t \geq 10$ days.
<b>CR</b>	Complete Response
<b>PR</b>	Partial Response
<b>SD</b>	Stable Disease
<b>PD</b>	Progressive Disease
<b>BRCA</b>	Breast invasive carcinoma
<b>CRC</b>	Colorectal carcinoma
<b>IC<sub>50</sub></b>	Half maximal inhibitory concentration
<b>MCC</b>	Matthews correlation coefficient
<b>MSE</b>	Mean squared error
<b>MAE</b>	Mean absolute error
<b>LOTO</b>	Leave one tissue out
<b>LOOCV</b>	Leave one out cross validation

## Declarations

### Acknowledgements

Thanks Yun Lu from Peking University and Yulong Pei from Eindhoven University of Technology for giving advices on the experiments and manuscript writing.

### Funding

Not applicable.

### Availability of data and materials

The NIBR-PDXE dataset is publicly available at <sup>[1]</sup>.

The CCLE and GDSC data can be downloaded from <sup>[2]</sup> and <sup>[3]</sup>.

The NCI-ALMANAC data can be downloaded from <sup>[4]</sup> and the preprocessing code is available at <sup>[5]</sup>.

The PPI data can be downloaded from <sup>[6]</sup>. The source code can be found from <sup>[7]</sup>.

### Authors' Contributions

YX prepared the datasets, conducted the experiments, and initially drafted the manuscript. YZ and JP conceived and guided the project. Both of them revised the manuscript.

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Competing Interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>Computer Science Department, Indiana University Bloomington, Bloomington, USA. <sup>2</sup>Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, USA. <sup>3</sup>Department of ISE, University of Illinois at Urbana-Champaign, Champaign, USA.

### References

1. Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A.A., Kim, S., Wilson, C.J., Lehár, J., Kryukov, G.V., Sonkin, D., *et al.*: The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**(7391), 603 (2012)
2. Yang, W., Soares, J., Greninger, P., Edelman, E.J., Lightfoot, H., Forbes, S., Bindal, N., Beare, D., Smith, J.A., Thompson, I.R., *et al.*: Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research* **41**(D1), 955–961 (2012)
3. Shoemaker, R.H.: The nci60 human tumour cell line anticancer drug screen. *Nature Reviews Cancer* **6**(10), 813 (2006)
4. Nguyen, L., Naulaerts, S., Bomane, A., Bruna, A., Ghislat, G., Ballester, P.: Machine learning models to predict in vivo drug response via optimal dimensionality reduction of tumour molecular profiles. *bioRxiv*, 277772 (2018)
5. Chiu, Y.-C., Chen, H.-I.H., Zhang, T., Zhang, S., Gorthi, A., Wang, L.-J., Huang, Y., Chen, Y.: Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC medical genomics* **12**(1), 18 (2019)
6. Xia, F., Shukla, M., Brettin, T., Garcia-Cardona, C., Cohn, J., Allen, J.E., Maslov, S., Holbeck, S.L., Doroshov, J.H., Evrard, Y.A., *et al.*: Predicting tumor cell line response to drug pairs with deep learning. *BMC bioinformatics* **19**(18), 486 (2018)
7. De Las Rivas, J., Fontanillo, C.: Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS computational biology* **6**(6), 1000807 (2010)
8. Ivanov, A.A., Khuri, F.R., Fu, H.: Targeting protein–protein interactions as an anticancer strategy. *Trends in pharmacological sciences* **34**(7), 393–400 (2013)
9. Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.F.: Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **18**(suppl.1), 233–240 (2002)

<sup>[1]</sup><http://www.nature.com/nm/journal/v21/n11/full/nm.3954.html#supplementary-information>.

<sup>[2]</sup><https://portals.broadinstitute.org/ccle/data>

<sup>[3]</sup>[https://www.cancerrxgene.org/downloads/bulk\\_download](https://www.cancerrxgene.org/downloads/bulk_download)

<sup>[4]</sup><https://wiki.nci.nih.gov/display/NCIDTPdata/NCI-ALMANAC>

<sup>[5]</sup><https://github.com/ECP-CANDLE/Benchmarks/tree/release.01/Pilot1/Combo>

<sup>[6]</sup><https://string-db.org/cgi/download.pl?sessionId=uKr0odAK9hPs>

<sup>[7]</sup><https://github.com/kinjad/GNNDR>

10. Ayati, M., Erten, S., Chance, M.R., Koyutürk, M.: Mobas: identification of disease-associated protein subnetworks using modularity-based scoring. *EURASIP Journal on Bioinformatics and Systems Biology* **2015**(1), 7 (2015)
11. Fischer, B., Sandmann, T., Horn, T., Billmann, M., Chaudhary, V., Huber, W., Boutros, M.: A map of directional genetic interactions in a metazoan cell. *eLife*, 4 (2015)
12. Gao, H., Korn, J.M., Ferretti, S., Monahan, J.E., Wang, Y., Singh, M., Zhang, C., Schnell, C., Yang, G., Zhang, Y., et al.: High-throughput screening using patient-derived tumor xenografts to predict clinical trial drug response. *Nature medicine* **21**(11), 1318 (2015)
13. Ali, M., Aittokallio, T.: Machine learning and feature selection for drug response prediction in precision oncology applications. *Biophysical reviews* **11**(1), 31–39 (2019)
14. Cortés-Ciriano, I., van Westen, G.J., Bouvier, G., Nilges, M., Overington, J.P., Bender, A., Malliavin, T.E.: Improved large-scale prediction of growth inhibition patterns using the nci60 cancer cell line panel. *Bioinformatics* **32**(1), 85–95 (2015)
15. Li, M., Wang, Y., Zheng, R., Shi, X., Wu, F., Wang, J., et al.: Deepdsc: A deep learning method to predict drug sensitivity of cancer cell lines. *IEEE/ACM transactions on computational biology and bioinformatics* (2019)
16. Wang, H., Liu, R., Schyman, P., Wallqvist, A.: Deep neural network models for predicting chemically induced liver toxicity endpoints from transcriptomic responses. *Frontiers in pharmacology* **10**, 42 (2019)
17. Vinayagam, A., Zirin, J., Roesel, C., Hu, Y., Yilmazel, B., Samsonova, A.A., Neumüller, R.A., Mohr, S.E., Perrimon, N.: Integrating protein-protein interaction networks with phenotypes reveals signs of interactions. *Nature methods* **11**(1), 94 (2014)
18. Coley, C.W., Jin, W., Rogers, L., Jamison, T.F., Jaakkola, T.S., Green, W.H., Barzilay, R., Jensen, K.F.: A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science* **10**(2), 370–377 (2019)
19. Han, P., Yang, P., Zhao, P., Shang, S., Liu, Y., Zhou, J., Gao, X., Kalnis, P.: Gcn-mf: Disease-gene association identification by graph convolutional networks and matrix factorization. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 705–713 (2019). ACM
20. Zitnik, M., Agrawal, M., Leskovec, J.: Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34**(13), 457–466 (2018)
21. Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Guerrero, R., Glocker, B., Rueckert, D.: Disease prediction using graph convolutional networks: Application to autism spectrum disorder and alzheimer's disease. *Medical image analysis* **48**, 117–130 (2018)
22. Song, T.-A., Chowdhury, S.R., Yang, F., Jacobs, H., El Fakhri, G., Li, Q., Johnson, K., Dutta, J.: Graph convolutional neural networks for alzheimer's disease classification. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 414–417 (2019). IEEE
23. Wee, C.-Y., Liu, C., Lee, A., Poh, J.S., Ji, H., Qiu, A., Initiative, A.D.N., et al.: Cortical graph neural network for ad and mci diagnosis and transfer learning across populations. *NeuroImage: Clinical* **23**, 101929 (2019)
24. Alshahrani, M., Hoehndorf, R.: Semantic disease gene embeddings (smudge): phenotype-based disease gene prioritization without phenotypes. *Bioinformatics* **34**(17), 901–907 (2018)
25. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
26. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems*, pp. 1024–1034 (2017)
27. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015)
28. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: *Advances in Neural Information Processing Systems*, pp. 4800–4810 (2018)
29. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: *Advances in Neural Information Processing Systems*, pp. 2204–2212 (2014)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017)
31. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014)
32. Lee, J., Lee, I., Kang, J.: Self-attention graph pooling. *arXiv preprint arXiv:1904.08082* (2019)
33. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., Jegelka, S.: Representation learning on graphs with jumping knowledge networks. *arXiv preprint arXiv:1806.03536* (2018)
34. Holbeck, S.L., Camalier, R., Crowell, J.A., Govindharajulu, J.P., Hollingshead, M., Anderson, L.W., Polley, E., Rubinstein, L., Srivastava, A., Wilsker, D., et al.: The national cancer institute almanac: A comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity. *Cancer research* **77**(13), 3564–3576 (2017)
35. Reinhold, W.C., Sunshine, M., Liu, H., Varma, S., Kohn, K.W., Morris, J., Doroshow, J., Pommier, Y.: Cellminer: a web-based suite of genomic and pharmacologic tools to explore transcript and drug patterns in the nci-60 cell line set. *Cancer research* **72**(14), 3499–3511 (2012)
36. Szklarczyk, D., Gable, A.L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N.T., Morris, J.H., Bork, P., et al.: String v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research* **47**(D1), 607–613 (2018)
37. Von Mering, C., Jensen, L.J., Snel, B., Hooper, S.D., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M.A., Bork, P.: String: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic acids research* **33**(suppl.1), 433–437 (2005)
38. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018)
39. Hahnloser, R.H., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S.: Digital selection and analogue

amplification coexist in a cortex-inspired silicon circuit. *Nature* **405**(6789), 947 (2000)

# Figures

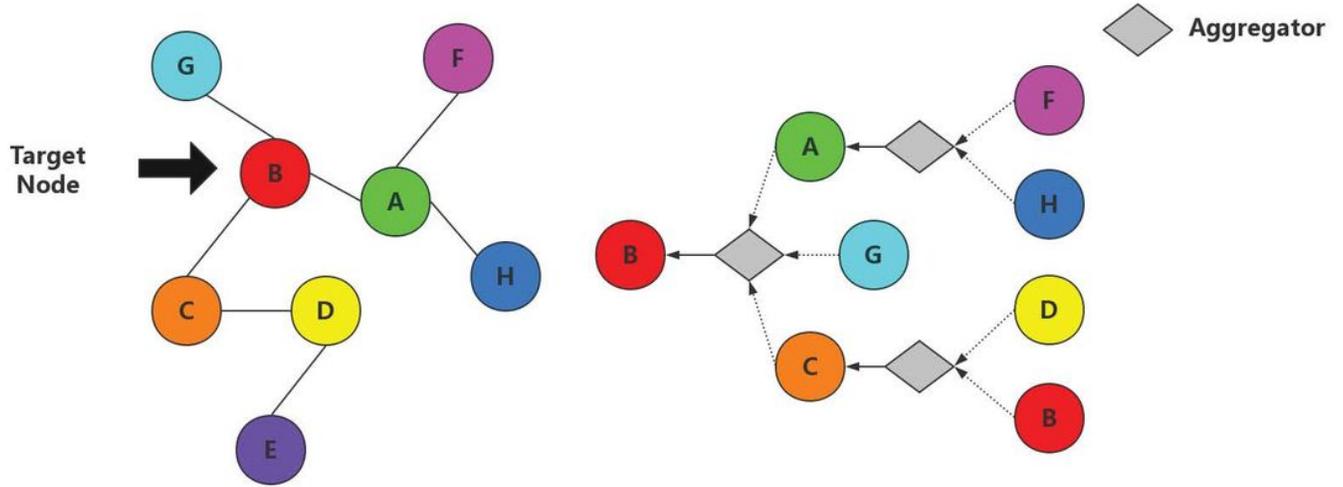
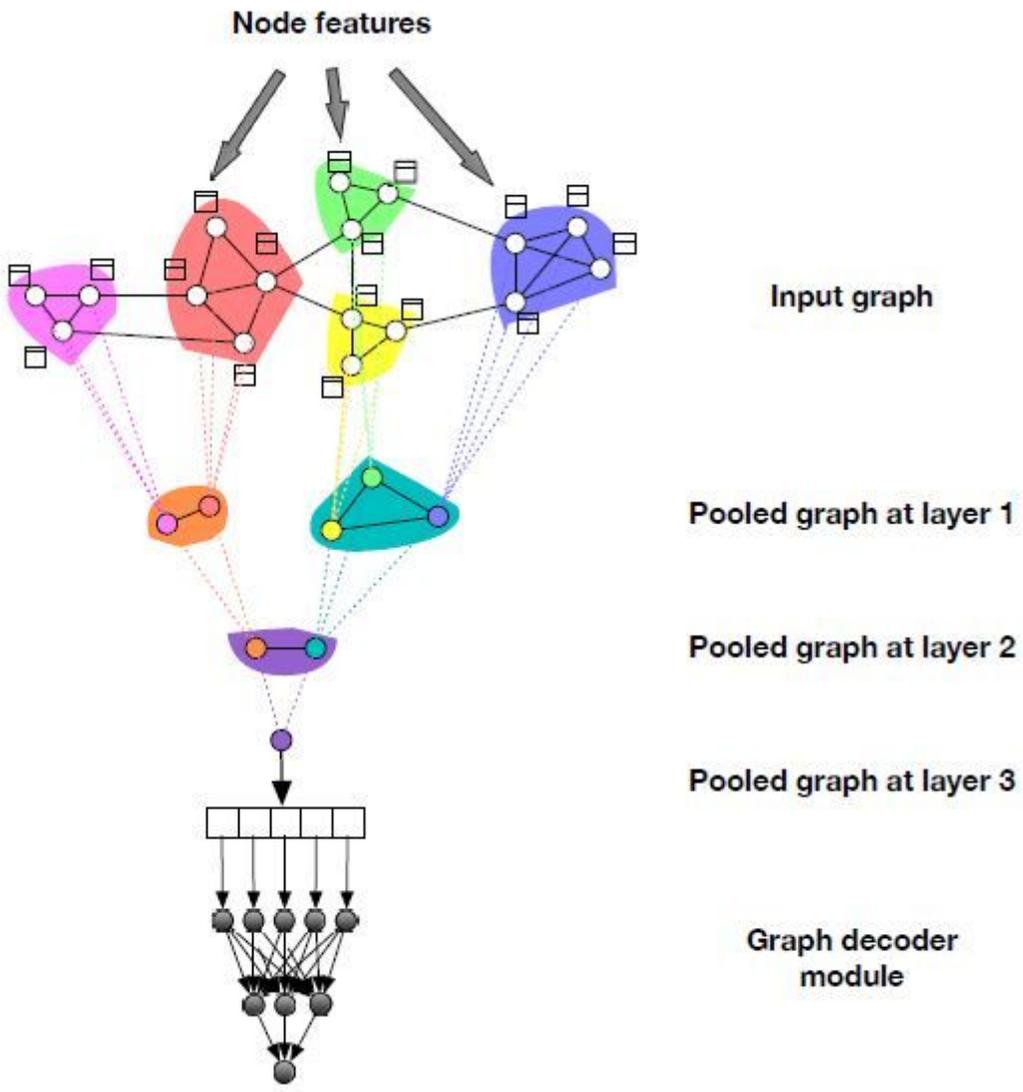


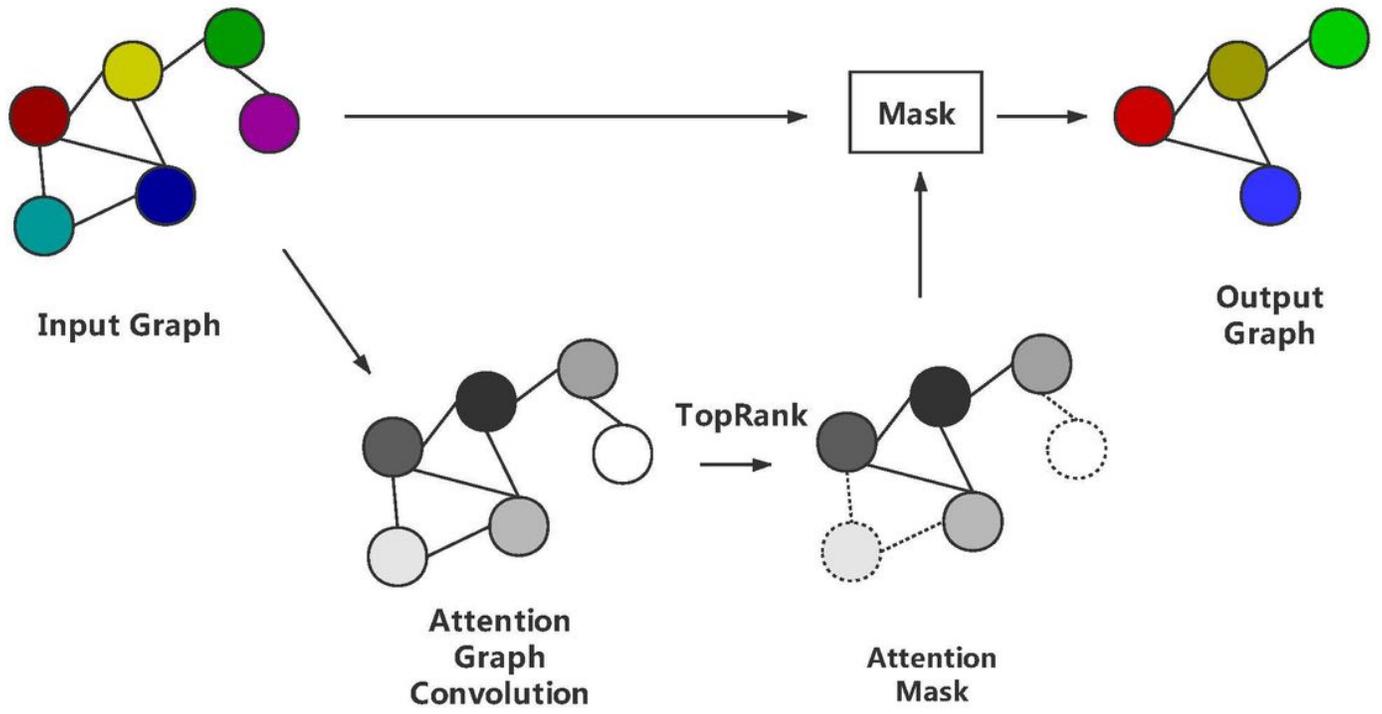
Figure 1

Illustration of graph convolution.



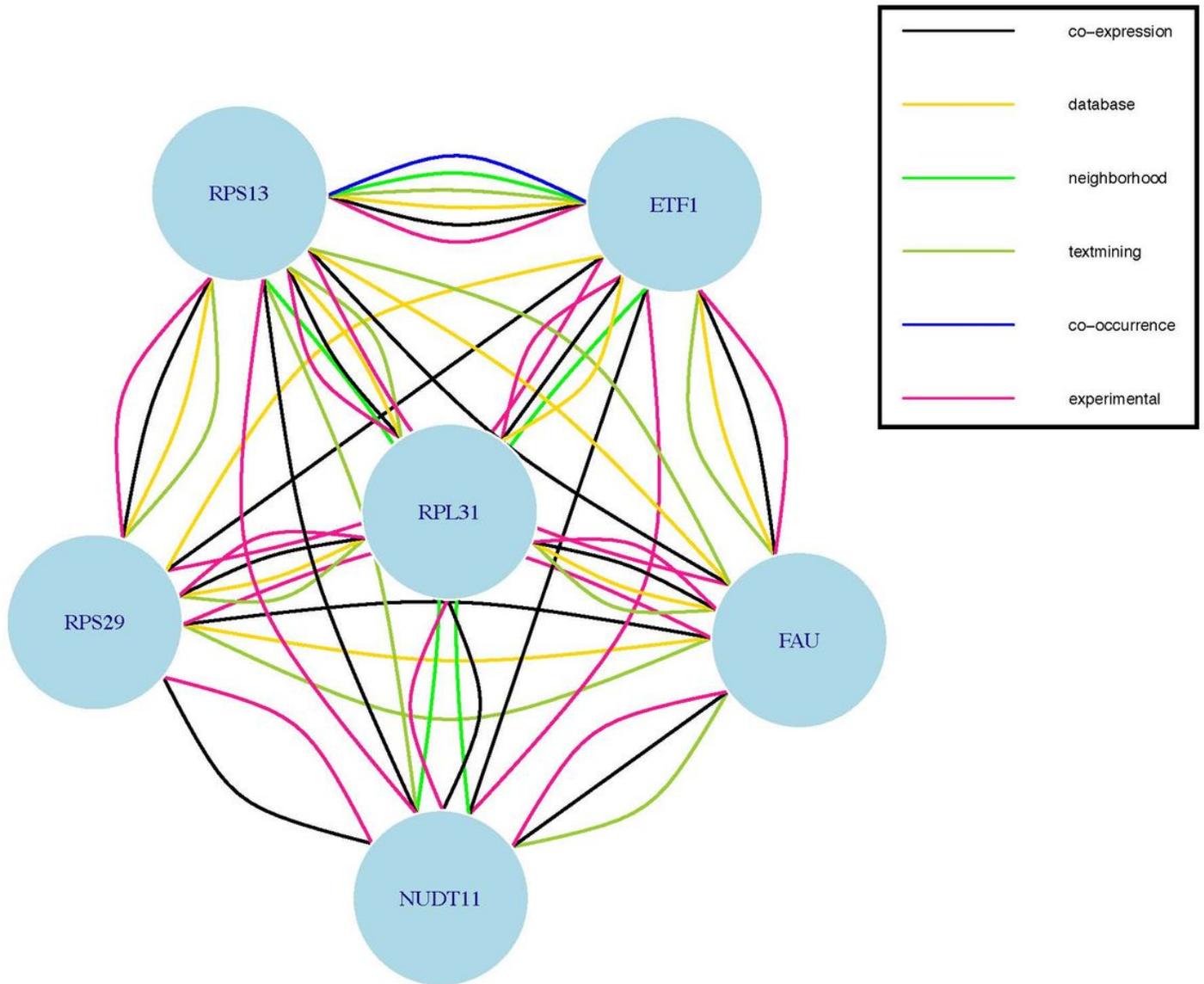
**Figure 2**

An illustration of hierarchical differential graph pooling.



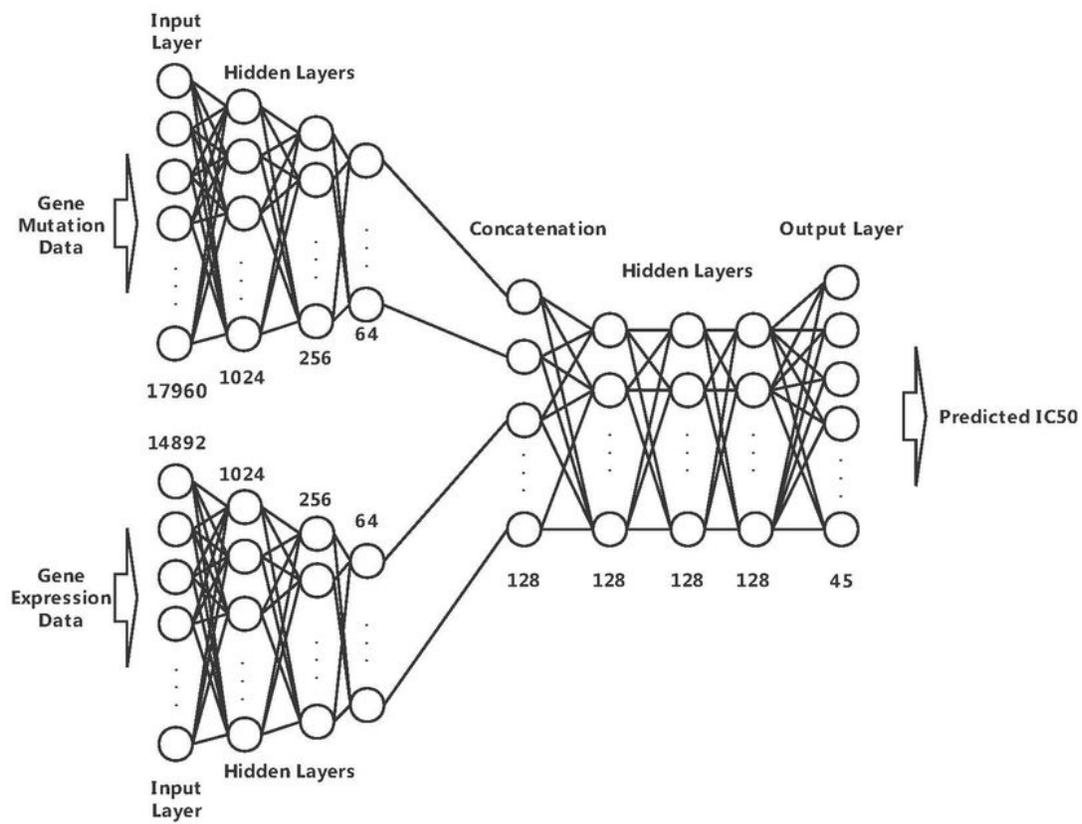
**Figure 3**

An illustration of the SAGpool layer.

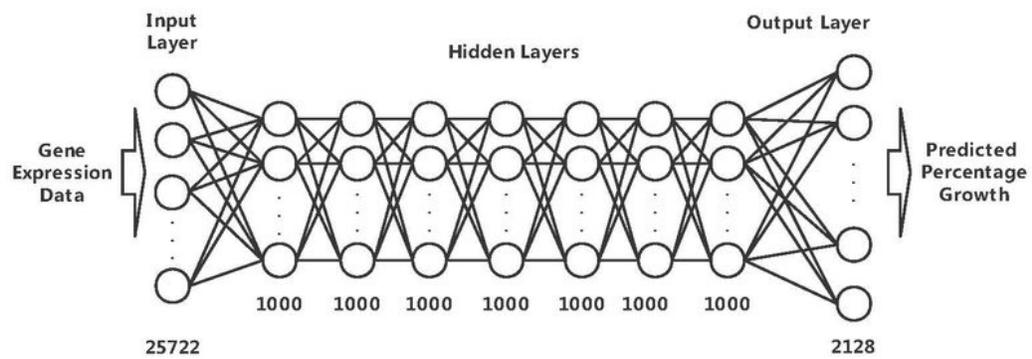


**Figure 4**

An example of interactions among proteins. Nodes denote proteins and edges of different colors represent interactions of different types.



(a) DNN architecture for CCLL-GDSC



(b) DNN architecture for NCI-ALMANAC

Figure 5

Architecture of DNN baselines.

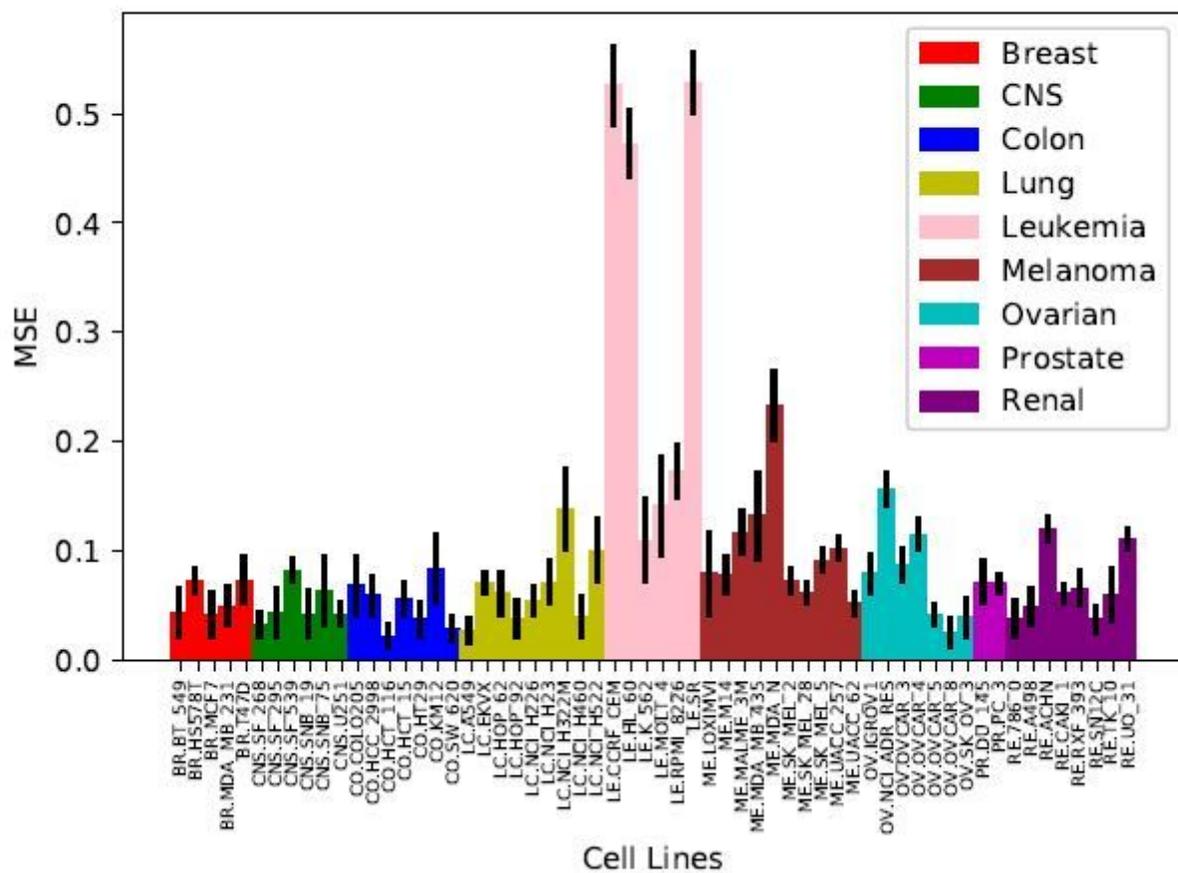


Figure 6

Prediction error from cell line view on NCI-ALMANAC.

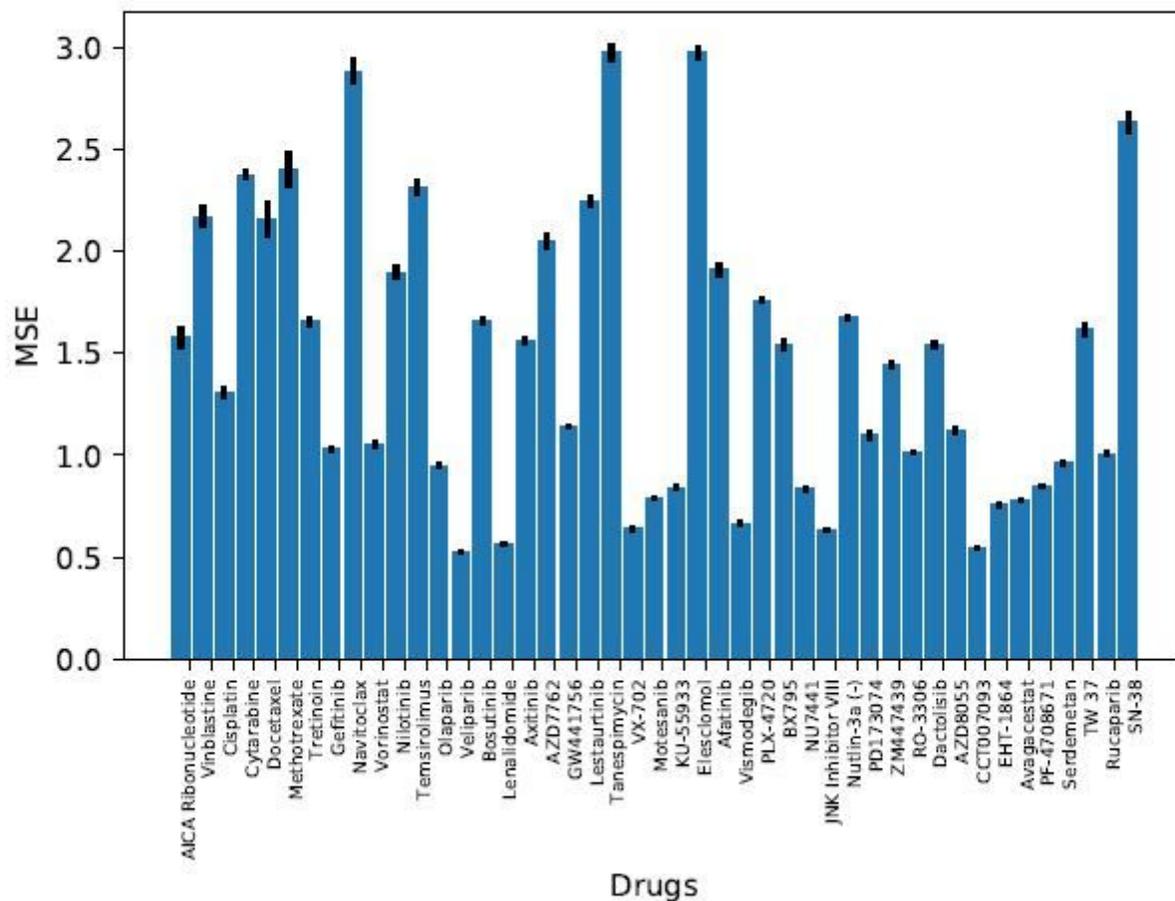
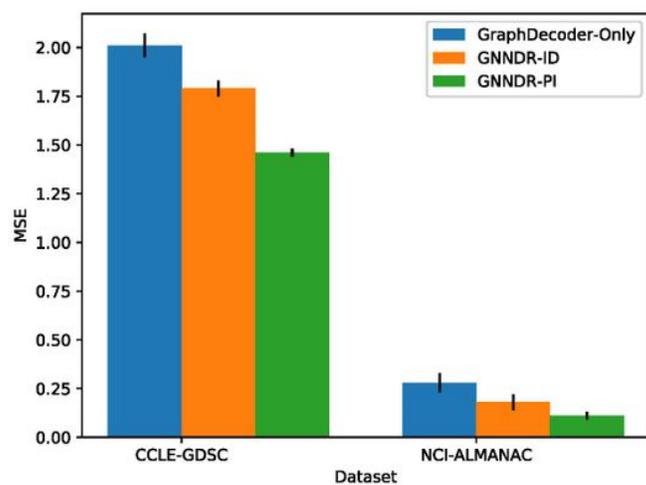
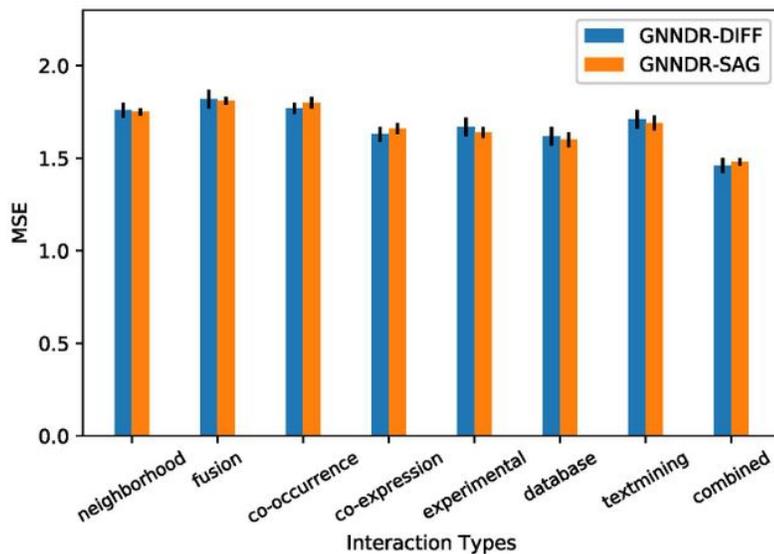


Figure 7

Prediction error from drug view on CCLE-GDSC.



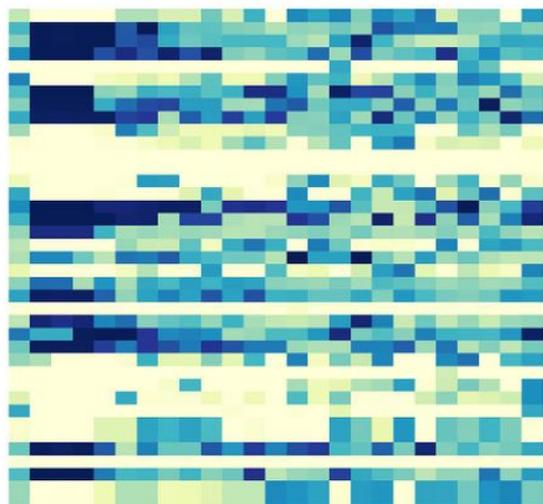
(a) Effectiveness of protein interaction



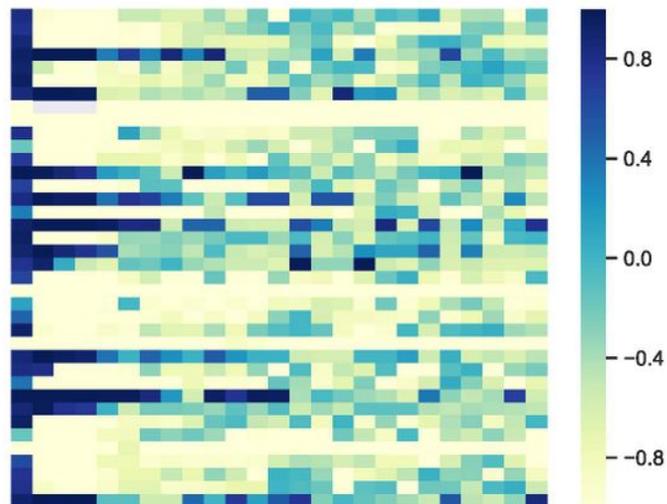
(b) Effectiveness of different types of protein interaction on CCLE-GDSC

## Figure 8

Effectiveness of PPI. (a) Compared to two baselines, incorporating PPIs indeed improves prediction performance. -ID stands for using identity matrix as adjacency matrix and -PI stands for using PPI network as adjacency matrix. (b) Impacts of different types of interactions.



(a) Graph attention scores on BRCA-binimetinib



(b) Graph attention scores on BRCA-paclitaxel

## Figure 9

Graph attention scores on BRCA-binimetinib and BRCA-paclitaxel. The rows denote the data points in each dataset and the columns represent genes which are listed in descending orders from left to right using correlation scores introduced in [4].