Prediction of Synthetic Lethal Interactions in Human Cancers using Multi-view Graph Auto-Encoder

Zhifeng Hao, Di Wu, Yuan Fang, Min Wu, Ruichu Cai, Xiaoli Li

***Abstract*—Synthetic lethality (SL) is a very important concept for the development of targeted anticancer drugs. However, experimental methods for SL detection often suffer from various issues like high cost and low consistency across cell lines. Hence, computational methods for predicting novel SLs have recently emerged as complements for wet-lab experiments. In addition, SL data can be represented as a graph where nodes are genes and edges are the SL interactions. It is thus motivated to design advanced graph-based machine learning algorithms for SL prediction. In this paper, we propose a novel SL prediction method using Multi-view Graph Auto-Encoder (SLMGAE). We consider the SL graph as the main view and the graphs from other data sources (e.g., PPI, GO, etc.) as support views. Multiple Graph Auto-Encoders (GAEs) are implemented to reconstruct the graphs for different views. We further design an attention mechanism, which assigns different weights for support views, to combine all the reconstructed graphs for SL prediction. The overall SLMGAE model is then trained by minimizing both the reconstruction error and prediction error. Experimental results on the SynLethDB dataset show that SLMGAE outperforms state-of-the-arts. The case studies on novel predicted SLs also illustrate the effectiveness of our SLMGAE method. The source codes, data, and supplementary materials for our SLMGAE are available via https://github.com/DiNg1011/SLMGAE.**

***Index Terms*—Synthetic lethality, graph neural network, graph auto-encoder, multi-view, human cancers.**

I. Introduction

Synthetic lethality, as an important concept for developing anti-cancer drug targets, has drawn great attention in the field of cancer therapeutics [1]. Specifically, a pair of genes form a synthetic lethality (SL) interaction if simultaneous defects of both genes result in cell death, while the defect of a single gene is not lethal. Given a gene with cancer-specific mutations, we can target its SL partners to selectively kill the cancer cells without harming normal cells [2]. Therefore, cancer therapeutics based on the SL concept can have less side effects compared with traditional chemotherapies [3]. Successful stories about the SL-based drugs include PARP-inhibitors Olaparib and Niraparib for ovarian and breast cancers [4]. These two drugs are based on the well-known SL interactions between genes PARP and BRCA1/BRCA2, as illustrated in Figure 1.

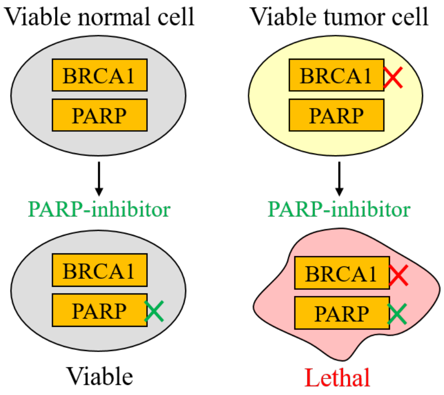


Fig. 1: PARP-inhibitor as an anti-cancer drug based on the SL interaction between PARP and BRCA1.

Wet-lab experiments have been developed to detect SL interactions. RNAi screening using siRNA or shRNA libraries can generate genome-wide SL data [5]. In addition, CRISPR-based genome editing technology can also be used for SL screening [6]. However, wet-lab experiments for SL detection have different challenges. For example, RNAi screening is lack of consistency across different cell lines and has off-target effects, while CRISPR-based genome editing technology is very expensive and also has off-target effects. Therefore, computational methods for human SL prediction have recently emerged as useful complements to the wet-lab experiments.

Graph neural network (GNN) is a powerful neural network architecture on graphs that can effectively capture the graph structures [7]. We are thus motivated to customize GNN to predict novel SLs in the SL graph. However, there remain some challenges using GNN for SL prediction. First, various data sources for genes (e.g., protein-protein interactions, gene ontology, etc.) would be useful for SL prediction. It is a challenge to integrate these data sources in a GNN-based framework for SL prediction. Second, different data sources may play different roles for SL prediction. Therefore, how to differentiate each data source would be another challenge in GNN-based SL prediction.

To address the above issues, we proposed a supervised

This work was partially supported by Natural Science Foundation of China (61876043), Natural Science Foundation of Guangdong (2014A030306004, 2014A030308008), Guangdong High-level Personnel of Special Support Program (2015TQ01X140), Science and Technology Planning Project of Guangzhou (201902010058). Yuan Fang, Min Wu and Ruichu Cai are corresponding authors.

Zhifeng Hao, Di Wu and Ruichu Cai are with the School of Computer, Guangdong University of Technology, Guangzhou, China, 510006 (e-mail: zfhao@fosu.edu.cn; diongvonbf@gmail.com; cairuichu@gmail.com).

Yuan Fang is with the School of Information Systems, Singapore Management University, Singapore, 178902 (e-mail: yfang@smu.edu.sg).

Min Wu and Xiaoli Li are with the Institute for Infocomm Research, A*STAR, Singapore, 138632 (e-mail: {wumin, xlli}@i2r.a-star.edu.sg).

multi-view graph auto-encoder denoted as SLMGAE to integrate various data sources for human SL prediction. First, we model the SL data and the other data sources (e.g., PPI, GO, etc.) as graphs. Second, we implement multiple Graph Auto-Encoders (GAEs) to reconstruct these graphs. Third, we further design an attentive merging process to combine all the reconstructed graphs for SL prediction. In particular, we derive the overall loss for both graph reconstruction and SL prediction and then train the SLMGAE model by minimizing this loss. We summarize our contributions as follows.

- We proposed a multi-view framework based on graph neural networks (i.e., graph auto-encoder) for human SL prediction.
- We designed an attention mechanism to effectively integrate the data from different views for SL prediction.
- Experimental results demonstrated that our proposed SLMGAE outperform the state-of-the-arts. Our qualitative case studies on novel predictions also illustrate the effectiveness of our proposed method.

The rest of the paper is organized as follows. Section II surveys the related works on SL prediction methods and graph neural networks. In Section III, we introduce the preliminaries for the SL prediction problem and describe the details of the SLMGAE model. Section IV presents the experimental setting and evaluation results. Finally, we conclude in Section V.

II. Related Work

In this section, we first introduce existing methods for human SL prediction. We also introduce the recent development of graph convolutional network techniques for bioinformatics applications.

A. SL Prediction Methods

Recently, various methods have been proposed for human SL prediction. We can divide these methods into two categories, namely, knowledge-based methods and supervised machine learning methods.

Knowledge-based methods utilize the knowledge or hypotheses to predict potential SL interactions. The hypotheses include that SL genes tend to (1) be co-expressed, (2) have similar functions, (3) have similar network properties, (4) exhibit mutual exclusivity with respect to specific genetic events [8], etc. For example, DAISY [9] predicts SLs from copy number variation data, gene expression data and shRNA data, based on the assumption that SL genes are often co-expressed and seldom co-mutated. MiSL [10] also predicts SLs by analyzing the data for mutation, copy number alternation and gene expression. Jacunski *et al.* [11] predicted human gene pairs with similar network parameters/characteristics (i.e. connectivity homology) to existing yeast SL pairs, as SL interactions. Pairs of genes that are altered in a mutually exclusive manner in cancers are observed to be likely to form SL interactions [8]. Knowledge-based methods are usually explainable for novel predictions. However, they do not explore the underlying patterns in the known human SL interactions.

Human SL data has recently been well curated in public databases, such as BioGRID [12] and SynLethDB [13]. Therefore, supervised machine learning techniques have been applied for human SL prediction. DiscoverSL [14], based on random forest classifier, predicts and visualizes novel human SLs using multi-omics cancer data (i.e., mutation, copy number alternation and gene expression data from TCGA) as features. SLant [15] extracts gene features from PPI and GO, and then predicts human SL using random forest classifier. Hence, DiscoverSL and SLant can be considered as traditional feature-based methods, which require to manually extract various gene features from different data sources. In fact, some feature-based methods, which are proposed for yeast SL prediction including MNMC [16] and MetaSL [17], can also be applied for human SL prediction. In addition, several matrix factorization methods, e.g., SL2MF [18], GRSMF [19] and CMF [**?**], have been proposed to predict human SL interactions. For example, SL2MF employs a logistic matrix factorization (LMF) based method and also integrates PPI and GO data for human SL prediction. Different from DiscoverSL and SLant, matrix factorization methods aim to automatically learn gene embeddings/features for SL prediction. However, matrix factorization, as a direct encoding [20], does not fully explore the graph structural information (e.g., neighborhoods) for human SL prediction.

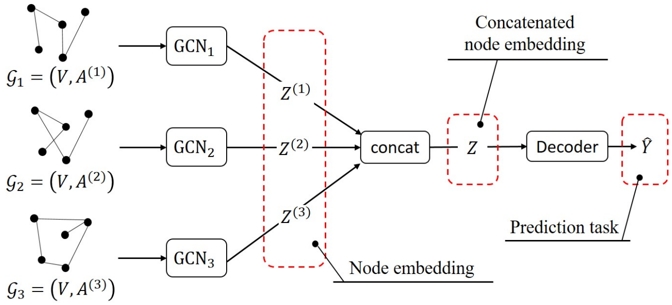


Fig. 2: Multi-view graph auto-encoder with GCN as encoder

B. Graph Neural networks

Graph Neural Networks (GNN) extend the existing neural networks for modeling the graph data [7]. GNN, such as graph convolutional network (GCN) and graph auto-encoder (GAE), has been used for drug discovery [**?**], disease prediction [21], microbe-drug association prediction [22], etc. Recently, a method called DDGCN [23] has been proposed for human SL prediction based on graph convolutional network. In particular, they employ both coarse-grained node dropout and fine-grained edge dropout to learn accurate gene embeddings for SL prediction. However, DDGCN predicts novel human SL pairs based solely on the known SL pairs and does not utilize any other data sources for genes.

Multi-view methods [24]–[26] have been proposed to integrate multiple data sources that are modelled as graphs. In particular, MVGCN [24] was proposed to fuse multiple modalities of brain images with multi-view graph convolutional network for Parkinson's disease prediction. As shown in Figure 2, MVGCN learns the node embedding from each individual view and then concatenates them to form the combined embedding for the prediction task. Similarly, an attention based multi-view graph auto-encoder was proposed for drug-drug interaction prediction [25]. However, so far there are no existing studies that use multi-view GNN to address the SL prediction task.

III. Methods

In this section, we first describe the notations and formulate the problem, then we introduce our SLMGAE model in details.

A. Preliminary

Based on the known SL interactions, we can construct a SL graph $\mathcal{G}^{SL} = (V, E)$. The nodes in $V = \{v_i\}_{i=1}^{n}$ are genes, where n is the total number of genes in the graph, and the edges in E are SL interactions. Moreover, $\mathcal{G}^{SL}$ is equivalent to an adjacency matrix $A^{SL} \in \mathbb{R}^{n \times n}$, where A^{SL} is 1 if (v_i, v_j) is a SL pair and 0 otherwise. We denote $\mathcal{G}^{SL}$ or A^{SL} as our *main view* for SL prediction.

We also utilize multi-omics data as additional inputs for SL prediction, e.g., gene ontology (GO) data, protein-protein interaction (PPI) data, etc. We denote these multi-omics data as *support views*. They can also be modelled as graphs, e.g., A^u is the graph for the support view u, $1 \leq u \leq n_s$ and n_s is the total number of support views.

In this paper, we leverage the data from both main view and support views for SL prediction. Specifically, we define $\mathcal{O}$ as the upper triangular array of A^{SL}. Furthermore, we denote the set of known SL interactions as $\mathcal{O}^+ = \{(v_i, v_j)|A_{ij} = 1, 1 \leqslant i < n, i < j \leqslant n\}$, and thus $\mathcal{O}^- = \mathcal{O}/\mathcal{O}^+$ is the set of unknown SL pairs. Our task is to build a model and predict the likelihood of gene pairs in $\mathcal{O}^-$ to form SL interactions. Table I lists the mathematical notation used in the paper.

B. Overview of SLMGAE

Figure 3 shows the overall framework of the proposed SLMGAE method.

First, we consider the SL graph as main view and the graphs constructed for other data sources (e.g., PPI, GO, etc.) as support views. Second, multiple GAEs are implemented to reconstruct the graphs from different views. In particular, GCN is used as the encoder to learn gene embeddings, while the decoder is used for graph reconstruction from the learned embeddings. We can then derive the reconstruction losses for both main view and support views. Third, we further design an attentive merging process to combine all the reconstructed graphs for SL prediction. Lastly, the overall SLMGAE model is then trained by minimizing both the reconstruction loss and prediction loss. Next we introduce each step of our SLMGAE model in details.

C. Graph reconstruction using GAE

As shown in Figure 3, GCN is used as the encoder in SLMGAE. Here, we use the standard propagation rule for the GCN [27], as shown in Equation 1.

TABLE I: List of notations

Symbol	Description
$\mathcal{G} = (V, E)$	gene interactions graph
$V = \{v_i\}_{i=1}^{n}$	nodes, i=1,...,N
$E = \{e_{ij}\}_{i,j=1}^{n}$	edges, i=1,...,N
$A \in \mathbb{R}^{n \times n}$	an adjacency matrix
$D \in \mathbb{R}^{n \times n}$	an degree matrix
A^{SL}	adjacency matrix of SL interactions
$\{A^u\}_{u=1}^{n_s}$	adjacency matrices of support views, u=1,...,n_s
$\mathcal{O}$	the upper triangular of A^{SL}
$\mathcal{O}^+$	$\{(v_i, v_j)\|A_{ij} = 1, 1 \leqslant i < n, i < j \leqslant n\}$
$\mathcal{O}^-$	$\{(v_i, v_j)\|A_{ij} = 0, 1 \leqslant i < n, i < j \leqslant n\}$
Z	node embedding matrix
F	initial node features
W	trainable weight matrix
$a^u \in \mathbb{R}^{n \times n}$	attention matrix, u=1,...,n_s
W_{supp}	support view weighted matrix
S	reconstruction graph
$(\cdot)^m$	superscript m denotes main view specific
$(\cdot)^u$	superscript u denotes support view specific, $u = 1, 2, ..., n_s$
$(\cdot)_l$	subscript l denotes l^{th}-layer GCN specific, $l = 1, 2, ...$
$(\cdot)_d$	subscript d denotes decoder specific
$\mathcal{L}_M$	main view reconstruction loss
$\mathcal{L}_S$	support view reconstruction loss
$\mathcal{L}_P$	SL prediction reconstruction loss

$$Z_{(l)} = \sigma(\hat{A} Z_{(l-1)} W_{(l)}), \tag{1}$$

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

$$\tilde{A} = I + A, \tag{3}$$

where $Z_{(l-1)}$ and $Z_{(l)}$ are the inputs and outputs for the l^{th}-layer, when $l = 1$, the input Z_0 of GCN is the initial node features F as shown in Figure 3. $W_{(l)}$ is a layer-specific trainable weight matrix, $\sigma(\cdot)$ denotes activation function (e.g., sigmoid or ReLU), $\tilde{D}$ is diagnonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $\tilde{A}$ is the graph adjacency matrix with self-loop.

In this paper, we use 2-layer GCN and we can thus compute the node embedding matrix Z with Equations 4 and 5. F in Equation 4 is the initial feature matrix (F is Z_0 as mentioned above), while Z_1 is the node embedding matrix learned in the first layer. Following DDGCN [23], we also take the SL graph adjacency matrix A^{SL} as node features F for all the graph convolutional networks (i.e., encoders) as shown in Figure 3. In addition, W_1 and W_2 are the weight matrices, and σ_1 and σ_2 are the activation functions in the 1st and 2nd GCN layers.

$$Z_1 = \sigma_1(\hat{A} F W_1), \tag{4}$$

$$Z_2 = \sigma_2(\hat{A} Z_1 W_2). \tag{5}$$

Using LeakyReLU as the activation function, we can compute the node embedding matrix Z_2^m for the main view in Equation 7. We further reconstruct the graph S^m for the main view using the weighted inner-product decoder in Equation 8, and W_d^m is a view-specific trainable matrix in the decoder.

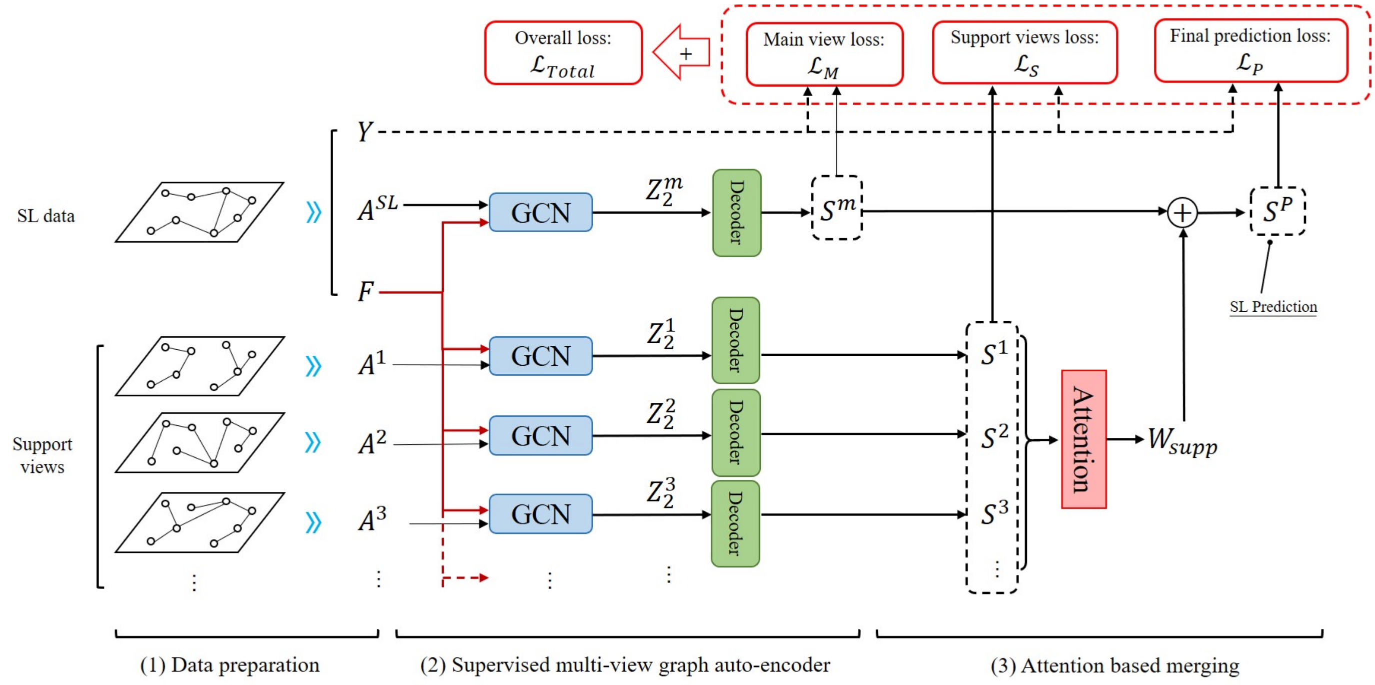


Fig. 3: The overall framework of our proposed SLMGAE method

$$Z_1^m = LeakyReLU(\hat{A}^{SL}FW_1^m), \tag{6}$$
$$Z_2^m = LeakyReLU(\hat{A}^{SL}Z_1^mW_2^m), \tag{7}$$
$$S^m = Z_2^mW_d^mZ_2^{m^{\mathsf{T}}}. \tag{8}$$

Here $LeakyReLU(\cdot)$ is defined as follows:

$$LeakyReLU(x) = max(0.2x, x). \tag{9}$$

Similarly, we can also compute the node embedding matrix Z_2^u and the reconstructed graph S^u for each support view u ($1 \leq u \leq n_s$, n_s is the number of support views) in Equations 11 and 12. As our eventual goal is to predict novel SL interactions, we need to link the support views with SL so that the learned node embeddings are useful for SL prediction. Therefore, we also use the SL matrix A^{SL} as the initial feature matrix F in Equation 10 for learning the node embeddings.

$$Z_1^u = LeakyReLU(\hat{A}^uFW_1^u), \tag{10}$$
$$Z_2^u = LeakyReLU(\hat{A}^uZ_1^uW_2^u), \tag{11}$$
$$S^u = Z_2^uW_d^uZ_2^{u^{\mathsf{T}}}. \tag{12}$$

Given a reconstructed graph S and its original graph Y, we define the reconstruction loss as their mean square error (MSE) in Equation 13.

$$MSE(Y,S) = \frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(Y_{ij} - S_{ij})^2, \tag{13}$$

We set Y as A^{SL}, and obtain S^m and S^u by Equations 8 and 12. We can thus derive the reconstruction loss $\mathcal{L}_M$ for the main view and $\mathcal{L}_S$ for the support views as follows.

$$\mathcal{L}_M = MSE(Y, S^m), \tag{14}$$
$$\mathcal{L}_S = \sum_u MSE(Y, S^u). \tag{15}$$

Note that our reconstructed graphs in Equations 8 and 12 are asymmetric matrices, while the SL matrix Y is symmetric. In order to avoid predicting two different values for the same SL pair, we only consider the upper triangle of the reconstructed matrices in Equation 13 when we calculate the above losses.

D. Attentive merging layer for SL prediction

In the section above, we have reconstructed the score matrices (or graphs) for both main view and support views. Next, we introduce the attentive merging layer, which merges the reconstructed score matrices for SL prediction.

First, we implement an edge-level attention mechanism to combine all the reconstructed score matrices from the support views. In particular, we randomly initialize a weight $a'^u_{ij} \in \mathbb{R}^{n\times n}$ ($1 \leq i, j \leq n$ and $1 \leq u \leq n_s$) for the node pair (v_i, v_j) under the view u. We then normalize the weight across different views using the softmax function in Equation 16. As such, we can obtain the normalized edge-level attention matrix $a^u \in \mathbb{R}^{n\times n}$ for the view u.

$$a^u_{ij} = \frac{e^{a'^u_{ij}}}{\sum\limits_{x=1}^{n_s} e^{a'^x_{ij}}} \tag{16}$$

Based on the above attention scheme, we can derive a weighted similarity matrix W_{supp} as Equation 17. In particular, $\circ$ is Hadamard product (i.e., the element-wise multiplication).

$$W_{supp} = \sum_{u}^{n_s} a^u \circ S^u \tag{17}$$

And then, we derive the final score matrix S^P by combining our main view matrix S^m and the weighted score matrix W_{supp} from the support views in Equation 18.

$$S^P = S^m + CW_{supp}, \tag{18}$$

where C is a hyper-parameter to control the contribution of W_{supp} for final prediction. We leverage the final score matrix S^P for SL prediction. Therefore, we can also use the mean square error to measure the final prediction loss $\mathcal{L}_P$ in Equation 19.

$$\mathcal{L}_P = MSE(Y, S^P) \tag{19}$$

E. Overall loss and optimization

We combine the reconstruction losses (i.e., $\mathcal{L}_M$ and $\mathcal{L}_S$) and the prediction loss $\mathcal{L}_P$ and obtain the the overall loss $\mathcal{L}_{Total}$ as follows.

$$\mathcal{L}_{Total} = \mathcal{L}_M + \alpha\mathcal{L}_S + \beta\mathcal{L}_P, \tag{20}$$

where α and β are hyper-parameters, controlling the contributions from $\mathcal{L}_S$ and $\mathcal{L}_P$, respectively.

Let Θ denotes all the trainable parameters and it includes GCN weight matrices and attention matrices. Our SLMGAE model is then trained by minimizing the overall loss $\mathcal{L}_{Total}(\Theta)$ as follows. We use the Adam optimizer [28] for optimization and the training process of our SLMGAE model is illustrated in Algorithm 1.

$$\arg\min_{\Theta} \mathcal{L}_{Total}(\Theta) \tag{21}$$

Algorithm 1 Proposed SLMGAE Model

```
Input: The adjacency matrix of known SL interaction $A^{SL}$, the adjacency matrix of each support view $A^{(u)}$, learning rate $\eta$, dropout rate $\gamma$, main view score weight $C$, and number of iterations $n\_iter$
Output: Predicted matrix $S^P$
Initialize parameter $\Theta$ randomly;
for $t = 1$ to $n\_iter$ do
    Compute $Z_2^m$ and $Z_2^u$ using Equations 7 and 11;
    Compute the score matrices $S^m$ and $S^u$ using Equations 8 and 12;
    Compute the weighted matrix for support views $W_{supp}$ as shown in Equation 17;
    Compute the final matrix $S^P$ using Equation 18;
    Compute total loss $\mathcal{L}_{Total}$ using Equation 20;
    Update the parameter $\Theta$ to minimize $\mathcal{L}_{Total}$ by Adam with learning rate $\eta$;
end for
return $S^P$.
```

IV. Results and Discussions

In this section, we first introduce the experimental setup. Then, we demonstrate the performance of our proposed SLMGAE. Last, we present the case studies and show the top SL pairs predicted by our method.

A. Experimental Setup

1) Data: We downloaded SynLethDB [13], the most comprehensive up-to-date database for human SL interactions, to evaluate the performance of our SLMGAE method. SynLethDB contains 19,667 human SL pairs among 6,375 genes. Therefore, the graph for this SL data is very sparse, with less than 0.1% of the elements in its adjacency matrix are known SL pairs. We also utilized various data sources as support views in SLMGAE, including GO and PPI. GO has three sub-ontologies, namely, biological process (BP), molecular function (MF), and cellular component (CC). We downloaded the latest version of ontology file from http://geneontology.org/, where we extracted 28,747 BP terms, 11,153 MF terms and 4,184 CC terms. Given two proteins, we calculated their functional similarity using the method proposed by Wang *et al.* [29]. As such, we can obtain a gene similarity matrix for each sub-ontology. After we derived the similarity matrices, we used the k-nearest neighbour algorithm to build the GO graphs. For each gene, we selected its top-k neighbors (i.e., the k genes with the highest similarities with the given gene) and discarded the other neighbors. Hence, we can build a GO graph for each sub-ontology. In our experiments, we used two GO graphs (i.e., BP and CC) as the support views. For PPI data, we downloaded the latest version of BioGRID [12]. We removed the SL interactions from BioGRID and obtained a PPI graph with 98,581 protein-protein interactions among 6,375 genes.

Following Liany *et al.* [**?**], we obtained the second SL dataset with 245 SL interactions related to breast cancer in SynLethDB. We denoted this sub-dataset for breast cancer as SynLethDB-BC. In particular, these 245 SL pairs involve 332 genes and thus the main view graph has 332 nodes. As for the support views, we adopted the same five data sources as Liany *et al.* [**?**], including co-expression, mutual exclusivity scores, pathway co-membership, protein complex co-membership and PPI scores from the Hippie database [30]. The matrices for all the five support views have the same number of genes as the main view (i.e., matrices for the main view and support views have the same dimension of 332×332).

To conduct independent validation experiments, we downloaded the Gene Interactions (GIs) data [31] as our third SL dataset, which was collected by CRISPR interference in 2 leukemia cancer cell lines “K562” and “Jurkat”. First, we extracted the gene pairs with GI scores less than -3 as positive SL pairs, while the gene pairs with GI scores close to 0 as negative pairs. Second, we used the data from K562 as the training set, and the data from Jurkat as the test data. We built the main view SL graph based on the 1,429 positive SL pairs in K562. Following the same setting for SynLethDB and SynLethDB-BC datasets, we leveraged these 1,429 positive SL pairs and all the unknown pairs in K562 to train the model, while the test data in Jurkat consists of 280 positive SL pairs and the same number of sampled negative pairs. Third, we also extracted the gene feature matrices from GO (i.e., BP and CC) and PPI as support views for this GIs data. The details of above three datasets SynLethDB, SynLethDB-BC and GIs are summarized in Table II.

TABLE II: Summary of three SL datasets

	SynLethDB	SynLethDB-BC	GIs data
# human genes	6,375	332	449
# SL pairs	19,677	245	1,719
Average degree	6.17	1.47	8.90
Density	0.097%	0.0044%	0.020%
# support views	3	5	3

2) Baselines: In our experiments, we mainly compared the methods for SL prediction by learning latent representations of genes, namely BLM-NII [32], SL2MF [18], GRSMF [19], CMF [**?**], GAE [33], DDGCN [23], MVGCN [24] and A-MVGAE [25]. We summarize the above baselines as follows.

- **SL2MF** was proposed for human SL prediction based on logistic matrix factorization (LMF). It can also integrate GO semantic similarities and PPI topological similarities between genes in the LMF framework.
- **GRSMF** was proposed for human SL prediction based on matrix factorization. It can leverage graph regularized from different source data to improve SL prediction.
- **CMF** was proposed by liany *et al.* [**?**], which used to integrate different data sources and learn gene representation through collective matrix factorization (CMF) for human SL prediction. We used both the variants CMFW and g-CMF as the baselines.
- **BLM-NII** was proposed for drug-target interaction (DTI) prediction and Liu *et al.* [18] customized it as a baseline for SL prediction.
- **GAE** was proposed for link prediction and Cai *et al.* [23] customized it as a baseline for SL prediction.
- **MVGCN** was proposed to fuse multiple modalities of brain images for Parkinson's Disease prediction [24]. We customized it for SL prediction.
- **A-MVGAE** was proposed for drug-drug interaction (DDI) prediction using multi-view graph auto-encoder with an attention mechanism [25]. In our experiments, we adopted it for SL prediction.
- **DDGCN** was a graph convolutional network framework with a dual-dropout mechanism, namely, coarse-grained node dropout and fine-grained edge dropout for human SL prediction.

3) Parameter setting: On SynLethDB dataset, we used a 2-layer GCN for our SLMGAE model and the dimensions of trainable weight matrix in the first and second layers were set to 512 and 256, respectively. The learning rate η, dropout rate γ, the parameters α, β and C were set to 0.001, 0.2, 2.0, 4.0 and 2.0. We adapted Adam optimizer to train our model for 300 epochs. In addition, we built KNN graphs for the support views and the k for both BP and CC were set to 45. We also empirically tuned the parameter setting for the baselines on SynLethDB dataset. In SL2MF, the parameters $c, \gamma, \alpha, \beta, \theta$ were set to $50, 2^{-5}, 2^{-1}, 2^{-2}, 2^{-1}$, respectively, where α, β and θ were the weights for the three support views. In GRSMF, the parameters λ and weight coefficient α, β, γ for each view were set to $2^7, 2^1, 2^{-1}, 2^7$. In CMFW, the dimension of latent representation k was set to 128. In BLM-NII, we set the value of the linear combination weight as 0.75 and used the max function to generate the prediction scores. GCN based methods including GAE, DDGCN, MVGCN and A-MVGAE have parameter settings similar to our model, which are summarized in Table III.

TABLE III: Parameter settings for GCN based methods

Parameters	SLMGAE	GAE	DDGCN	MVGCN	A-MVGAE
Learning rate η	0.001	0.01	0.01	0.001	0.001
Dropout rate γ	0.2	0.3	0.5	0.4	0.3
# training epochs	300	2,000	2,000	300	300
early stop threshold	–	1e-5	1e-5	–	–
# GCN layers	2	2	2	2	2
# units in layer1	512	512	512	512	512
# units in layer2	256	256	256	256	256

On SynLethDB-BC and GIs datasets, most of the parameter settings were the same as those on SynLethDB dataset. Considering that the above two datasets is much smaller than SynLethDB, we modified some parameters in our SLMGAE and baselines. For example, we reduced the number of units from 512 to 128 in layer-1 and 256 to 64 in layer-2 respectively. Please refer to our supplementary materials for more details about parameter setting on SynLethDB-BC and GIs datasets.

4) Evaluation metrics: For the experiments on SynLethDB and SynLethDB-BC, we conducted 5-fold cross-validation to evaluate various models for SL prediction. In particular, we equally split the known SL interactions into five non-overlapping subsets. We iteratively chose one subset as positive samples and sampled the same number of unknown pairs as negative samples. We put these positive and negative samples together for testing. The remaining unknown pairs and four subsets of known SL pairs were used for training.

We used three metrics for performance evaluation, including the area under ROC(receiver operating characteristic) curve (AUROC), the area under Precision-Recall curve (AUPR) and the best F1 score achievable on the Precision-Recall curve.

For ROC curve, the true positive rate (TPR) and the false positive rate (FPR) are first defined in Equation 22.

$$\begin{aligned} TPR = TP/(TP+FN), \\ FPR = FP/(FP+TN), \end{aligned} \tag{22}$$

where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. The ROC curve is then created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, and AUROC is the area under the ROC curve.

Similarly, the Precision and Recall are defined in Equation 23 (Recall is the same as true positive rate, TPR, in Equation 22). Then, the Precision-Recall curve (PRC) is created to show the trade-off between Precision and Recall at various threshold settings, and AUPR is the area under the Precision-Recall curve.

$$\begin{aligned} Recall = TP/(TP+FN), \\ Precision = TP/(TP+FP). \end{aligned} \tag{23}$$

Lastly, F1 score is defined as the harmonic mean of precision and recall. In this work, we have adopted the best value of F1 score achieved on the Precision-Recall curve. In addition, SL prediction is a typical Positive and Unlabeled (PU) learning task. Therefore, we also used a modified version of F1 score for performance evaluation under PU setting [34], [35]. Detailed results in terms of the modified F1 score can be found in our supplementary materials.

B. Results on SynLethDB and SynLethDB-BC

We show the performance comparison among various methods for SL prediction on both SynLethDB and SynLethDB-BC datasets in this section.

Table IV shows the performance comparison among various methods on SynLethDB in terms of AUROC, AUPR and F1. Based on the results in Table IV, we can have the following three observations. First, SLMGAE clearly outperforms the matrix factorization methods including SL2MF, CMF and GRSMF. Considering that they all used additional data sources (e.g., GO and PPI), SLMGAE's superior performance demonstrates the advantage of GCN-based method over matrix factorization methods. Second, SLMGAE, which integrates multiple support views, also achieves higher performance than GCN-based methods including GAE and DDGCN. Third, SLMGAE outperforms multi-view GCN-based methods (i.e., MVGCN and A-MVGAE) indicates that our model can aggregate information from multiple perspectives more effectively. In particular, DDGCN as the state-of-the-art method prior to this work achieves the second best AUPR as shown in Table IV, while SLMGAE further achieves improvements over DDGCN by 4.45%, 3.02% and 6.25% in terms of AUROC, AUPR and F1, respectively.

Figure 4 shows the ROC and PRC for the various methods. In order to make the picture more readable, we only plotted the performance curves for six methods in the figure, including SLMGAE, SL2MF, GAE, DDGCN, CMFW and A-MVGAE. For more comparison on various datasets, please refer to our supplementary materials.

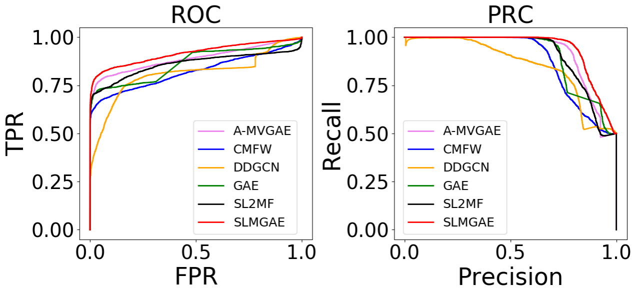


Fig. 4: ROC and PRC of various methods on SynLethDB

Table IV also shows the performance of various methods on the SynLethDB-BC dataset. We can observe that our SLMGAE model still achieves superior performance on this SynLethDB-BC dataset. In particular, SLMGAE achieves the highest AUROC and F1 scores, and the second best AUPR (slightly lower than SL2MF). In addition, we noticed that g-CMF performed very well on the small-scale dataset (SynLethDB-BC). However, it was not generalized well to the larger datasets and achieved poor performance on SynLethDB. Although we adjusted the model parameters, we still observed insignificant changes in model loss during training g-CMF on SynLethDB, leading to a poor performance on this larger dataset.

C. Independent validation on GIs dataset

Table V shows the results of independent validation performed on the GIs data (i.e., trained on K562 cell line and tested on Jurkat cell line). From Table V it can be observed that our model performs better than all baseline algorithms in terms of three metrics. It can also be observed that the multi-view GCN-based methods (i.e., SLMGAE, MVGCN, and A-MVGAE) outperforms the single-view GCN-based methods (i.e., GAE and DDGCN), indicating that additional data sources can be helpful for SL prediction. In conclusion, independent validation experiments on the GIs data also illustrate the effectiveness of our method.

D. Comparison with feature-based methods on SynLethDB

We also compared our method with 10 state-of-the-art feature-based classification methods on SynLethDB. They include K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Naive Bayesian (NB), AdaBoost, GradientBoost, Bootstrap aggregating (Bagging), MNMC [16] and MetaSL [17]. We extracted 18 features from various data sources, e.g., GO, PPI and etc. Please refer to our supplementary materials for more details about features. On SynLethDB dataset with 6,375 genes, we have over 20,000,000 unknown pairs as negatives to train our SLMGAE model. Hence, the number of negative samples is too big to run traditional feature-based methods (such as SVM and Random Forest). Therefore, we adopted a new experimental setting so that we can compare our model with traditional feature-based methods. In particular, we sampled the same numbers of negative samples as positives for both training and testing (i.e., 1:1 positive and negative pairs). For example, we conducted 5-fold cross validation on SynLethDB dataset with 19,677 positive SL pairs. We first sampled 19,677 unknown pairs as negative samples. We divided the positive and negative samples into 5 groups, where 1 group for testing and the remaining 4 groups for training.

Table VI shows the performance of various feature-based methods on the SynLethDB dataset under 1:1 positive and negative pairs training. From Table VI, we can observe that our model outperforms all feature-based methods. Compared to the best performing feature-based method (i.e., MetaSL), our method also showed significant superiority, with improvements of 5.40%, 5.64% and 9.98% in AUROC, AUPR and F1 respectively.

To summarize, we can conclude from the above comparisons that our SLMGAE model outperforms the state-of-the-art methods (representation learning methods and traditional feature-based methods) on both the large and small datasets. Thereafter, we will show the results of the model ablation study and parameter analysis only on the larger dataset, i.e., SynLethDB dataset.

TABLE IV: Performance comparison of various SL prediction methods under 5-fold cross-validation

Methods	SynLethDB			SynLethDB-BC		
	AUROC	AUPR	F1	AUROC	AUPR	F1
BLM-NII	0.6116 ± 0.0157	0.6507 ± 0.0281	0.6319 ± 0.0469	0.8692 ± 0.0559	0.8261 ± 0.0746	0.8665 ± 0.0688
SL2MF	0.8631 ± 0.0053	0.9106 ± 0.0026	0.8176 ± 0.0053	0.8920 ± 0.0424	**0.9310** ± **0.0293**	0.8716 ± 0.0497
CMFW	0.8209 ± 0.0030	0.8798 ± 0.0017	0.7795 ± 0.0021	0.6174 ± 0.0527	0.7327 ± 0.0615	0.7056 ± 0.0272
g-CMF	0.5536 ± 0.0057	0.5646 ± 0.0057	0.6469 ± 0.0049	0.9043 ± 0.0423	0.9228 ± 0.0381	0.8968 ± 0.0529
GRSMF	0.8642 ± 0.0046	0.8989 ± 0.0039	0.8220 ± 0.0032	0.7577 ± 0.0468	0.6967 ± 0.0693	0.7671 ± 0.0416
GAE	0.8664 ± 0.0043	0.9028 ± 0.0045	0.8307 ± 0.0065	0.8653 ± 0.0708	0.8989 ± 0.0476	0.8926 ± 0.0524
DDGCN	0.8783 ± 0.0040	0.9152 ± 0.0022	0.8204 ± 0.0048	0.8713 ± 0.0563	0.9256 ± 0.0280	0.9112 ± 0.0359
MVGCN	0.8556 ± 0.0049	0.9036 ± 0.0046	0.8326 ± 0.0039	0.9103 ± 0.0337	0.9251 ± 0.0342	0.8818 ± 0.0226
A-MVGAE	0.8796 ± 0.0036	0.9128 ± 0.0029	0.8489 ± 0.0048	0.9106 ± 0.0469	0.9291 ± 0.0533	0.8819 ± 0.0455
SLMGAE	**0.9174** ± **0.0045**	**0.9428** ± **0.0032**	**0.8717** ± **0.0074**	**0.9192** ± **0.0599**	0.9279 ± 0.0377	**0.9231**± **0.0490**

TABLE V: Independent validation results on GIs dataset

Methods	AUROC	AUPR	F1
BLM-NII	0.6971	0.7039	0.6809
SL2MF	0.6154	0.6765	0.6534
GRSMF	0.6884	0.7344	0.7254
CMFW	0.6609	0.7113	0.6675
g-CMF	0.5939	0.6279	0.5934
GAE	0.6031	0.6563	0.6468
DDGCN	0.6089	0.6589	0.6613
MVGCN	0.7202	0.7465	0.7038
A-MVGAE	0.7172	0.7347	0.6901
SLMGAE	**0.7834**	**0.7984**	**0.7372**

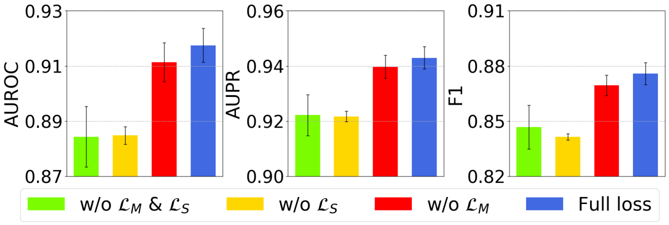


Fig. 5: Loss ablation study for our SLMGAE

TABLE VI: Comparison with feature-based methods on SynLethDB under 1:1 positive and negative training pairs

Methods	AUROC	AUPR	F1
RF	0.8536 ± 0.0049	0.8810 ± 0.0043	0.7766 ± 0.0048
DT	0.7277 ± 0.0034	0.7959 ± 0.0024	0.7309 ± 0.0028
NB	0.7278 ± 0.0038	0.7542 ± 0.0039	0.6893 ± 0.0028
SVM	0.7578 ± 0.0076	0.7840 ± 0.0083	0.7007 ± 0.0045
KNN	0.7256 ± 0.0031	0.7543 ± 0.0048	0.6908 ± 0.0020
Bagging	0.8514 ± 0.0042	0.8796 ± 0.0034	0.7753 ± 0.0030
AdaBoost	0.7987 ± 0.0024	0.8274 ± 0.0040	0.7271 ± 0.0031
GradientBoost	0.8378 ± 0.0032	0.8654 ± 0.0041	0.7595 ± 0.0043
MNMC	0.8279 ± 0.0024	0.8396 ± 0.0026	0.7600 ± 0.0023
MetaSL	0.8704 ± 0.0032	0.8931 ± 0.0016	0.7948 ± 0.0045
SLMGAE	**0.9174** ± **0.0041**	**0.9434** ± **0.0023**	**0.8741** ± **0.0046**

E. Model Ablation Study

Recall that we have three loss components in our SLMGAE model as shown in Figure 3, namely main view loss $\mathcal{L}_M$, support view loss $\mathcal{L}_S$ and the prediction loss $\mathcal{L}_P$. In this section, we conducted ablation study to investigate the impact of each individual loss. Specifically, we evaluated the performance of variants (1) without both $\mathcal{L}_M$ and $\mathcal{L}_S$ (i.e., with only $\mathcal{L}_P$); (2) without $\mathcal{L}_S$; (3) without $\mathcal{L}_M$; and (4) full loss (i.e., our SLMGAE model).

Figure 5 shows the performance of different variants. We can clearly observe that two variants without $\mathcal{L}_S$ achieve significantly lower scores, indicating that the support view loss is critical in our SLMGAE for SL prediction. In addition, the variant without $\mathcal{L}_M$ can still achieve relatively good performance (i.e., good AUROC and F1). The reason behind is that our final score matrix in Equation 18 is somehow determined by the matrix from the main view, and thus the two losses $\mathcal{L}_M$ and $\mathcal{L}_P$ have overlaps. However, $\mathcal{L}_M$ is still important as the full loss can achieve higher AUPR and make the model more stable as shown in Figure 5.

Note that our SLMGAE integrates three support views, namely BP, CC and PPI, for SL prediction. Here, we also examined the contribution of each support view. As shown in Figure 6, the main view together with any support view can achieve higher scores than without support views, demonstrating that support views can enhance the performance of SL prediction. Moreover, the attention mechanism can further integrate multiple support views in an effective manner. As such, our SLMGAE with all the support views can achieve the best performance as shown in Figure 6.

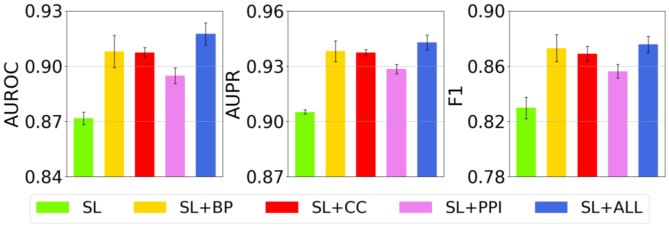
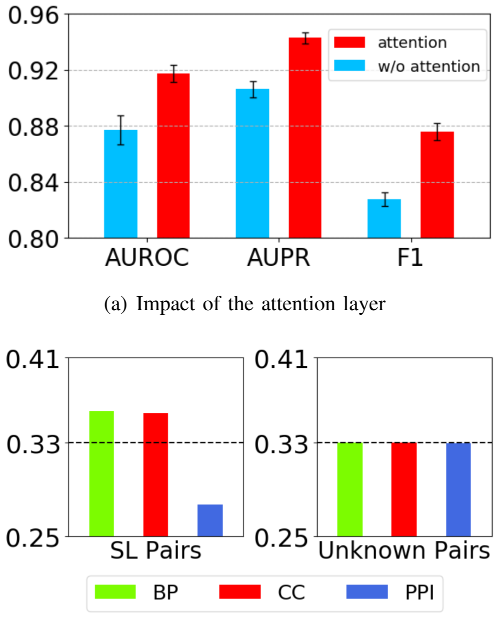


Fig. 6: Support view ablation study for our SLMGAE

Moreover, we show the impact of the designed attention mechanism in Figure 7. Firstly, Figure 7(a) shows the performance of our SLMGAE with or without the designed attention layer. We assigned equal weights for support views when we removed the attention layer from our SLMGAE model. As shown in Figure 7, the designed attention mechanism can help SLMGAE to achieve higher performance in terms of all the three metrics. Secondly, we averaged the attention scores for all the known SL pairs and unknown pairs in each view as shown in Figure 7(b). We can observe that BP and CC views are assigned with higher attention scores for known SL pairs than the PPI view. Such attention scores indicate that functional similarities based on GO information are more important than PPI data for SL prediction.

As mentioned above, SL prediction can also be treated as a Positive and Unlabeled (PU) learning problem [36], [37],



(a) Impact of the attention layer

(b) Average attention for each view

Fig. 7: Impact of the designed attention mechanism

which can be solved by minimising some ranking losses. We conducted experiments using a ranking loss [36], [37] in Equation 24 instead of MSE in our loss function, where $\mathcal{F}(\cdot) = \log(1+\exp(\cdot))$, $\mathcal{O}^+$ and $\mathcal{O}^-$ denote the positive and unknown pairs respectively. We denote the model using this ranking loss as SLMGAE-R.

$$RL(S) = \frac{1}{|\mathcal{O}^+|}\sum_{e^+\in\mathcal{O}^+}\mathcal{F}\left(\frac{1}{|\mathcal{O}^-|}\sum_{e^-\in\mathcal{O}^-}S_{e^-} - S_{e^+}\right), \quad (24)$$

Table VII shows the comparison between the ranking loss (SLMGAE-R) and the MSE loss (SLMGAE). We can observe that both losses work well for the SL prediction task. In particular, MSE loss performs better on the large-scale SynLethDB dataset, while the ranking loss performs slightly better on the small-scale SynLethDB-BC dataset.

TABLE VII: Comparison between ranking loss and MSE loss.

Methods	SynLethDB		
	AUROC	AUPR	F1
SLMGAE-R	0.9100 ± 0.0048	0.9281 ± 0.0044	0.8579 ± 0.0065
SLMGAE	**0.9174 ± 0.0045**	**0.9428 ± 0.0032**	**0.8717 ± 0.0074**
Methods	SynLethDB-BC		
	AUROC	AUPR	F1
SLMGAE-R	**0.9236 ± 0.0276**	**0.9358 ± 0.0167**	0.9191 ± 0.0261
SLMGAE	0.9192 ± 0.0599	0.9279 ± 0.0377	**0.9231± 0.0490**

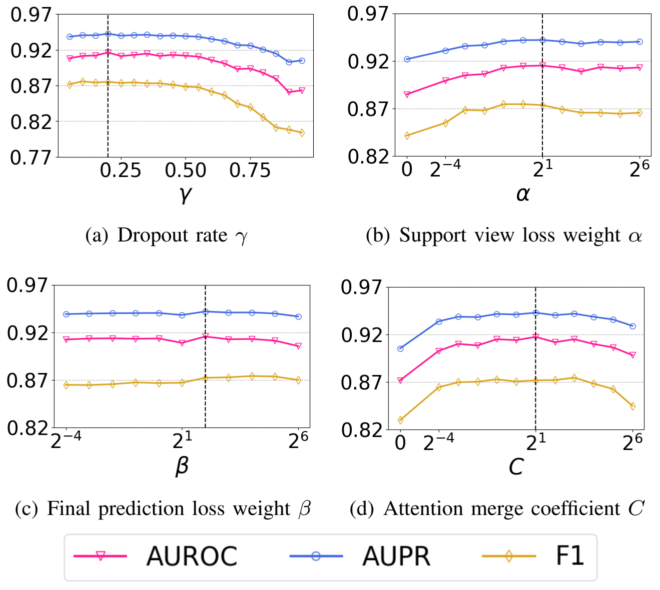
F. Parameter Sensitivity Analysis

In this section, we performed the sensitivity analysis for the dropout probability γ, two loss coefficients α and β and the parameter C.

Figure 8(a) shows the performance of SLMGAE with different dropout rates. We can observe that the performance of SLMGAE is relatively stable when $\gamma < 0.5$ and it becomes unstable (scores keep decreasing) when $\gamma > 0.6$. Overall, we would recommend to set γ in the range $[0.1, 0.4]$ and we reported the performance of SLMGAE by setting γ as 0.2.

α and β control the contribution of losses $\mathcal{L}_S$ and $\mathcal{L}_P$, respectively. In Figure 8(b), we tune α while β is fixed as 4. We can observe that the performance of SLMGAE is poor when $\alpha = 0$ (i.e., w/o L_S in Figure 5), and both AUPR and F1 are optimal when α is 2. In Figure 8(c) with α fixed as 2.0, AUROC and AUPR of SLMGAE are stable when we increase β, while F1 slightly increases. Overall, SLMGAE achieves relatively good performance when $\beta \in$ [4, 16] as shown in Figure 8(c). Eventually, we set α as 2 and β as 4 in our experiments.

Parameter C controls the contribution of W_{supp} in Equation 18 for final prediction. Figure 8(d) shows the performance of SLMGAE with different values for C. It is clear that the performance of SLMGAE is poor when $C = 0$, and then the performance remains stable when $C \in [2^{-1}, 2^3]$. In our experiments, we set the parameter C as 2.



(a) Dropout rate γ

(b) Support view loss weight α

(c) Final prediction loss weight β

(d) Attention merge coefficient C

Fig. 8: Sensitivity analysis for γ, α, β and C in SLMGAE

G. Case Studies for Top Predicted SL Pairs

We further trained our SLMGAE model using all the known SL pairs in SynLethDB and applied SLMGAE to predict novel SL interactions. We ranked these unknown pairs based on their scores predicted by SLMGAE, and subsequently searched for those top-ranked pairs in biomedical literature.

Among the top 1000 predicted SL pairs, we found 30 pairs supported by existing publications. In particular, 13 out of these 30 pairs have been validated by wet-lab experiments (e.g., CRISPR screening, siRNA screening, etc) and the remaining 17 pairs have been predicted by in-silico methods. More details about these 30 predicted SLs can be found in Table S4 in our supplementary materials. We further select 10 from the above 30 predicted SL pairs as shown in Table VIII. Column 2 and column 3 are two genes of the predicted SL pairs. Column 4 provides the PubMed ID of the publications supporting our predictions, while the last column shows the specific evidence for each predicted SL.

TABLE VIII: Top predicted SL pairs with literature support

#	Gene 1	Gene 2	PubMed ID	Evidence
1	BRCA1	KRAS	24104479	shRNA screening
2	CCT5	KRAS	28700943	CRISPR and shRNA screens
3	BRCA1	PIK3CA	26427375	in-silico prediction
4	HDAC9	MYC	29764852	HDAC and BRD inhibition
5	SRC	TP53	23728082	in-silico prediction
6	KRAS	UNC13B	24104479	shRNA screening
7	BRAF	TP53	24025726	in-silico prediction
8	KRAS	TRIP11	25407795	Combinatorial RNAi
9	SMAD3	TP53	23728082	in-silico prediction
10	AKT1	CHEK1	28319113	CRISPR-Cas9

As shown in Table VIII, several pairs have been verified by wet-lab experiments. KRAS is one of the most frequently mutated oncogenes in human cancers and many SL pairs involving KRAS have been detected by different techniques. Particularly, row 1 (BRCA1 and KRAS) and row 6 (KRAS and UNC13B) are validated by shRNA screening [38], row 2 (CCT5 and KRAS) is validated by combined CRISPR and shRNA sceens [39] and row 8 (KRAS and TRIP11) is detected by Combinatorial RNAi [40]. In addition, row 4 (HDAC9 and MYC) is verified by combined HDAC and bromodomain protein (BRD) inhibition [41] while row 10 (AKT1 and CHEK1) is verified by CRISPR-Cas9 [42].

Meanwhile, 4 SL pairs predicted by our SLMGAE in Table VIII are also predicted by other in-silico methods. For example, row 3 (BRCA1 and IK3CA) is also predicted in [8] based on the mutual exclusivity information. Row 5 (SRC and TP53) and row 9 (SMAD3 and TP53) are predicted by a network centrality-based method in [43], while row 7 (BRAF and TP53) is pre-screened in [44] using gene expression profiles. Given that these methods predict SL pairs based on the principles different from our SLMGAE method, we would consider that they still provide strong supports for our predictions.

V. Conclusion

SL interactions are important for cancer therapy and the computational methods for SL prediction can further provide potential targets in drug development for cancers. In this paper, we propose a Multi-view Graph Auto-Encoder based model named SLMGAE for predicting novel SL interactions. Experimental results show that our model outperforms other graph neural network methods and matrix factorization methods. Case studies also demonstrate that our proposed model is promising for predicting novel SLs.

Currently, our SLMGAE selects random gene-pairs as negative samples for model training and testing, which may have bias. In the future, we plan to extract negative SL pairs from DepMap [45] (https://depmap.org/), where the co-dependencies between genes are evaluated based on their effects in different cell lines of various cancers. With these meaningful negative SL pairs, we can thus generate unbiased decision boundary for better SL prediction. We also plan to integrate the gene knowledge graph with graph convolutional networks and graph attention networks for SL prediction. With various gene relationships in the knowledge graph, we aim to explain the underlying interacting mechanisms for predicted SL pairs. In addition, we will also work on the drug response screen data [46] to validate the predicted SL pairs in the future.

References

[1] J. D. Iglehart and D. P. Silver, “Synthetic lethality–a new direction in cancer-drug development,” New England Journal of Medicine, vol. 361, no. 2, p. 189, 2009.

[2] D. P. McLornan, A. List, and G. J. Mufti, “Applying synthetic lethality for the selective targeting of cancer,” New England Journal of Medicine, vol. 371, no. 18, pp. 1725–1735, 2014.

[3] N. J. O’Neil, M. L. Bailey, and P. Hieter, “Synthetic lethality and cancer,” Nature Reviews Genetics, vol. 18, no. 10, p. 613, 2017.

[4] D. A. Chan and A. J. Giaccia, “Harnessing synthetic lethal interactions in anticancer drug discovery,” Nature reviews Drug discovery, vol. 10, no. 5, p. 351, 2011.

[5] J. Luo, M. J. Emanuele, D. Li, C. J. Creighton, M. R. Schlabach, T. F. Westbrook, K.-K. Wong, and S. J. Elledge, “A genome-wide rnai screen identifies multiple synthetic lethal interactions with the ras oncogene,” Cell, vol. 137, no. 5, pp. 835–848, 2009.

[6] K. Han, E. E. Jeng, G. T. Hess, D. W. Morgens, A. Li, and M. C. Bassik, “Synergistic drug combinations for cancer identified in a crispr screen for pairwise genetic interactions,” Nature biotechnology, vol. 35, no. 5, p. 463, 2017.

[7] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2008.

[8] S. Srihari, J. Singla, L. Wong, and M. A. Ragan, “Inferring synthetic lethal interactions from mutual exclusivity of genetic events in cancer,” Biology direct, vol. 10, no. 1, p. 57, 2015.

[9] L. Jerby-Arnon, N. Pfetzer, Y. Y. Waldman, L. McGarry, D. James, E. Shanks, B. Seashore-Ludlow, A. Weinstock, T. Geiger, P. A. Clemons, et al., “Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality,” Cell, vol. 158, no. 5, pp. 1199–1209, 2014.

[10] S. Sinha, D. Thomas, S. Chan, Y. Gao, D. Brunen, D. Torabi, A. Reinisch, D. Hernandez, A. Chan, E. B. Rankin, et al., “Systematic discovery of mutation-specific synthetic lethals by mining pan-cancer human primary tumor data,” Nature communications, vol. 8, p. 15580, 2017.

[11] A. Jacunski, S. J. Dixon, and N. P. Tatonetti, “Connectivity homology enables inter-species network models of synthetic lethality,” PLoS computational biology, vol. 11, no. 10, p. e1004506, 2015.

[12] A. Chatraryamontri, R. Oughtred, L. Boucher, J. M. Rust, C. S. Chang, N. Kolas, L. Odonnell, S. Oster, C. L. Theesfeld, A. Sellam, et al., “The biogrid interaction database: 2017 update,” Nucleic Acids Research, vol. 45, 2017.

[13] J. Guo, H. Liu, and J. Zheng, “Synlethdb: synthetic lethality database toward discovery of selective and sensitive anticancer drug targets,” Nucleic acids research, vol. 44, no. D1, pp. D1011–D1017, 2015.

[14] S. Das, X. Deng, K. Camphausen, and U. Shankavaram, “Discoversl: an r package for multi-omic data driven prediction of synthetic lethality in cancers,” Bioinformatics, vol. 35, no. 4, pp. 701–702, 2018.

[15] G. Benstead-Hume, X. Chen, S. R. Hopkins, K. A. Lane, J. A. Downs, and F. M. Pearl, “Predicting synthetic lethal interactions using conserved patterns in protein interaction networks,” PLoS computational biology, vol. 15, no. 4, p. e1006888, 2019.

[16] G. Pandey, B. Zhang, A. N. Chang, C. L. Myers, J. Zhu, V. Kumar, and E. E. Schadt, “An integrative multi-network and multi-classifier approach to predict genetic interactions,” PLoS computational biology, vol. 6, no. 9, p. e1000928, 2010.

[17] M. Wu, X. Li, F. Zhang, X. Li, C.-K. Kwoh, and J. Zheng, “In silico prediction of synthetic lethality by meta-analysis of genetic interactions, functions, and pathways in yeast and human cancer,” Cancer informatics, vol. 13, pp. CIN–S14026, 2014.

[18] Y. Liu, M. Wu, C. Liu, X.-L. Li, and J. Zheng, “Sl 2 mf: Predicting synthetic lethality in human cancers via logistic matrix factorization,” IEEE/ACM transactions on computational biology and bioinformatics, vol. 17, no. 3, pp. 748–757, 2019.

[19] J. Huang, M. Wu, F. Lu, L. Ou-Yang, and Z. Zhu, “Predicting synthetic lethal interactions in human cancers using graph regularized self-representative matrix factorization,” BMC bioinformatics, vol. 20, no. 19, pp. 1–8, 2019.

[20] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," IEEE Data Eng. Bull., vol. 40, no. 3, pp. 52–74, 2017.

[21] Z. Sun, H. Yin, H. Chen, T. Chen, L. Cui, and F. Yang, "Disease prediction via graph neural networks," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 3, pp. 818–826, 2021.

[22] Y. Long, M. Wu, C. K. Kwoh, J. Luo, and X. Li, "Predicting human microbe–drug associations via graph convolutional network with conditional random field," Bioinformatics, vol. 36, no. 19, pp. 4918–4927, 2020.

[23] R. Cai, X. Chen, Y. Fang, M. Wu, and Y. Hao, "Dual-dropout graph convolutional network for predicting synthetic lethality in human cancers," Bioinformatics, vol. 36, no. 16, pp. 4458–4465, 2020.

[24] X. Zhang, L. He, K. Chen, Y. Luo, J. Zhou, and F. Wang, "Multi-view graph convolutional network and its applications on neuroimage analysis for parkinson's disease," in AMIA Annual Symposium Proceedings, vol. 2018, p. 1147, American Medical Informatics Association, 2018.

[25] T. Ma, C. Xiao, J. Zhou, and F. Wang, "Drug similarity integration through attentive multi-view graph auto-encoders," in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden (J. Lang, ed.), pp. 3477–3483, ijcai.org, 2018.

[26] S. K. Ata, Y. Fang, M. Wu, J. Shi, C. K. Kwoh, and X. Li, "Multi-view collaborative network embedding," ACM Trans. Knowl. Discov. Data, vol. 15, pp. 1–18, Apr. 2021.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (Y. Bengio and Y. LeCun, eds.), 2015.

[29] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen, "A new method to measure the semantic similarity of go terms," Bioinformatics, vol. 23, no. 10, pp. 1274–1281, 2007.

[30] G. Alanis-Lobato, M. A. Andrade-Navarro, and M. H. Schaefer, "Hippie v2. 0: enhancing meaningfulness and reliability of protein–protein interaction networks," Nucleic acids research, p. gkw985, 2016.

[31] M. A. Horlbeck, A. Xu, M. Wang, N. K. Bennett, C. Y. Park, D. Bogdanoff, B. Adamson, E. D. Chow, M. Kampmann, T. R. Peterson, et al., "Mapping the genetic landscape of human cells," Cell, vol. 174, no. 4, pp. 953–967, 2018.

[32] J.-P. Mei, C.-K. Kwoh, P. Yang, X.-L. Li, and J. Zheng, "Drug–target interaction prediction by learning from local information and neighbors," Bioinformatics, vol. 29, no. 2, pp. 238–245, 2012.

[33] T. N. Kipf and M. Welling, "Variational graph auto-encoders," arXiv preprint arXiv:1611.07308, 2016.

[34] W. S. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in the Twentieth International Conference on Machine Learning (ICML 2003), August 21-24, 2003, Washington, DC, USA, pp. 448–455, AAAI Press, 2003.

[35] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," Machine Learning, vol. 109, no. 4, pp. 719–760, 2020.

[36] I. M. Baytas, C. Xiao, F. Wang, A. K. Jain, and J. Zhou, "Heterogeneous hyper-network embedding," in IEEE International Conference on Data Mining (ICDM), pp. 875–880, 2018.

[37] Y. Liu, S. Qiu, P. Zhang, P. Gong, F. Wang, G. Xue, and J. Ye, "Computational drug discovery with dyadic positive-unlabeled learning," in Proceedings of the 2017 SIAM international conference on data mining, pp. 45–53, SIAM, 2017.

[38] F. J. Vizeacoumar, R. Arnold, F. S. Vizeacoumar, M. Chandrashekhar, A. Buzina, J. T. Young, J. H. Kwan, A. Sayad, P. Mero, S. Lawo, et al., "A negative genetic interaction map in isogenic cancer cell lines reveals cancer cell vulnerabilities," Molecular systems biology, vol. 9, no. 1, p. 696, 2013.

[39] T. D. Martin, D. R. Cook, M. Y. Choi, M. Z. Li, K. M. Haigis, and S. J. Elledge, "A role for mitochondrial translation in promotion of viability in k-ras mutant cells," Cell reports, vol. 20, no. 2, pp. 427–438, 2017.

[40] X. Wang, A. Q. Fu, M. E. McNerney, and K. P. White, "Widespread genetic epistasis among cancer genes," Nature communications, vol. 5, no. 1, pp. 1–10, 2014.

[41] Y. Zhang, C. T. Ishida, W. Ishida, S.-F. L. Lo, J. Zhao, C. Shu, E. Bianchetti, G. Kleiner, M. J. Sanchez-Quintero, C. M. Quinzii, et al., "Combined hdac and bromodomain protein inhibition reprograms tumor cell metabolism and elicits synthetic lethality in glioblastoma," Clinical Cancer Research, vol. 24, no. 16, pp. 3941–3954, 2018.

[42] J. P. Shen, D. Zhao, R. Sasik, J. Luebeck, A. Birmingham, A. Bojorquez-Gomez, K. Licon, K. Klepper, D. Pekin, A. N. Beckett, et al., "Combinatorial crispr–cas9 screens for de novo mapping of genetic interactions," Nature methods, vol. 14, no. 6, pp. 573–576, 2017.

[43] T. Kranthi, S. Rao, and P. Manimaran, "Identification of synthetic lethal pairs in biological systems through network information centrality," Molecular bioSystems, vol. 9, no. 8, pp. 2163–2167, 2013.

[44] X. Wang and R. Simon, "Identification of potential synthetic lethal genes to p53 using a computational biology approach," BMC medical genomics, vol. 6, no. 1, p. 30, 2013.

[45] A. Tsherniak, F. Vazquez, P. G. Montgomery, B. A. Weir, G. Kryukov, G. S. Cowley, S. Gill, W. F. Harrington, S. Pantel, J. M. Krill-Burger, et al., "Defining a cancer dependency map," Cell, vol. 170, no. 3, pp. 564–576, 2017.

[46] J. S. Lee, A. Das, L. Jerby-Arnon, R. Arafeh, N. Auslander, M. Davidson, L. McGarry, D. James, A. Amzallag, S. G. Park, et al., "Harnessing synthetic lethality to predict the response to cancer treatment," Nature communications, vol. 9, no. 1, pp. 1–12, 2018.