

Data and text mining

SLGNN: synthetic lethality prediction in human cancers based on factor-aware knowledge graph neural network

Yan Zhu¹, Yuhuan Zhou¹, Yang Liu ^{1,2}, Xuan Wang^{1,2} and Junyi Li ^{1,2,*}

¹School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China and ²Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China

*To whom correspondence should be addressed.

Associate Editor: Prof. Alfonso Valencia

Received on May 9, 2022; revised on November 29, 2022; editorial decision on January 9, 2023

Abstract

Motivation: Synthetic lethality (SL) is a form of genetic interaction that can selectively kill cancer cells without damaging normal cells. Exploiting this mechanism is gaining popularity in the field of targeted cancer therapy and anti-cancer drug development. Due to the limitations of identifying SL interactions from laboratory experiments, an increasing number of research groups are devising computational prediction methods to guide the discovery of potential SL pairs. Although existing methods have attempted to capture the underlying mechanisms of SL interactions, methods that have a deeper understanding of and attempt to explain SL mechanisms still need to be developed.

Results: In this work, we propose a novel SL prediction method, SLGNN. This method is based on the following assumption: SL interactions are caused by different molecular events or biological processes, which we define as SL-related factors that lead to SL interactions. SLGNN, apart from identifying SL interaction pairs, also models the preferences of genes for different SL-related factors, making the results more interpretable for biologists and clinicians. SLGNN consists of three steps: first, we model the combinations of relationships in the gene-related knowledge graph as the SL-related factors. Next, we derive initial embeddings of genes through an explicit message aggregation process of the knowledge graph. Finally, we derive the final gene embeddings through an SL graph, constructed using known SL gene pairs, utilizing factor-based message aggregation. At this stage, a supervised end-to-end training model is used for SL interaction prediction. Based on experimental results, the proposed SLGNN model outperforms all current state-of-the-art SL prediction methods and provides better interpretability.

Availability and implementation: SLGNN is freely available at <https://github.com/zy972014452/SLGNN>.

Contact: lijunyi@hit.edu.cn

1 Introduction

Developing effective anticancer drugs is a high priority research topic at medical and health research institutions worldwide. Identifying biochemical pathways that can be therapeutically targeted is a key step in developing new compounds. Synthetic lethality (SL) is an interaction between genes that, when both genes are mutated, can lead to reduced cell viability or even cell death, while mutations in only one of the genes in the pair not lethal (Guo, 2016). SL can selectively treat cancer cells by identifying existing mutations and targeting their synthetic lethal partners (Zhang, 2021). Detecting suitable lethal partners conventionally relies on high-throughput experimental laboratory screening technology, commonly using RNA interference (RNAi) (Luo, 2009) followed by genome editing through CRISPR (Du *et al.*, 2017). However, such approaches have severe limitations. For example, RNAi screening is

prone to off-target effects increasing the risk of clinical use (Topatana, 2020). In addition, the high cost, and relatively long-time scale limits the practical use of laboratory experiment-based screening of SL interactions.

To overcome these limitations, computational approaches are gaining considerable interest. Current computational methods can be subdivided into two categories: knowledge-based methods, and supervised machine learning (Long *et al.*, 2021). Knowledge-based approaches use prior knowledge, or assumptions, to predict SL gene pairs, mainly based on observations on known SL gene pairs. For example, DAISY (Jerby-Arnon *et al.*, 2014) used a data-driven approach to identify SL interactions within the genome, based on co-expressed but not co-mutated properties of SL gene pairs. Sinha *et al.* (2017) used genomics data from multiple sources to predict SL interactions, including mutations and copy number variations. While effective, the use of this approach is clearly limited by the

need for prior information related to SL, such as insight into biological metabolic networks, gene regulation, or other known interactions between previously recognized SL gene pairs. Clearly, such strategy is suboptimal in predicting novel SL interaction.

Advances in machine-learning-based methods resulted in its successful use in tasks, such as drug repurposing and gene–disease correlation in bioinformatics. Encouraged by these results, machine-learning methods have also been applied to predict SL interactions. The method proposed by Paladugu *et al.* (2008) obtained gene features from protein–protein interaction (PPI) network and used the extracted features to train a support vector machine classifier for SL interaction prediction. Das *et al.* (2019) predicted SL interactions using multi-omics data from the Cancer Genome Atlas database using a random forest model. Subsequently, Benstead-Hume *et al.* (2019) utilized gene features derived from PPI network and the Gene Ontology database to train a random forest model for SL interaction prediction. These methods are primarily based on traditional machine-learning processes. First, the features of genes are collated from different sources of data, and then the prediction model of SL interaction is derived by machine learning. As an alternative, there are methods based on graph representation learning. These methods model SL interactions as a graph, where the gene of interest is represented as a central node and potential SL interaction partners are located at the periphery. Graph representation learning relies on an encoder–decoder paradigm, in which the encoder propagates information between nodes through the topology of the graph to obtain low-dimensional representations of node features. The decoder part uses these features to complete downstream tasks like node classification, link prediction and community discovery (Hamilton, 2020). Depending on the implementation of this concept, there are two different forms of graph representation learning: matrix factorization (MF)-based methods and graph neural network (GNN)-based methods. Huang *et al.* (2019) proposed one such method, GRSMF, that used a self-representative MF encoder to predict SL interactions. Another approach, SL²MF (Liu *et al.*, 2020), obtained the features of genes by factorized differential gene similarity matrices from various sources and used these to predict potential SL interactions. Liany *et al.* (2020) used collective matrix factorization (CMF) to model multiple matrices for SL interaction prediction. A drawback of MF-based methods is that they simply generate a feature for each node, without parameter sharing or using node features, effectively a form of shallow embedding (Hamilton, 2020). Alternative approaches using GNNs can effectively alleviate this limitation. Cai *et al.* (2020) modeled SL interactions into a graph and used a novel dual-drop GNN to solve the sparsity problem of interaction networks. Long *et al.* (2021) proposed a novel graph contextualized attention network, GCATSL that predicted SL interactions by aggregating gene feature graphs from different sources based on an attention mechanism.

The above GNN- and MF-based methods attempt to capture the underlying biological mechanisms of SL by modeling the similarity of gene nodes. However, this capture process needs to be improved as it has limitations in the expressive capacity of homogeneous graphs. To mitigate the impact of this problem, Wang *et al.* (2021) proposed a knowledge graph-based approach, KG4SL, and was the first to apply this in biomedicine for the prediction of SL interactions. This method utilizes the rich semantic information present in the knowledge graph to capture the underlying mechanism of SL. While KG4SL uses the underlying mechanisms of SL interaction to enhance gene embedding it does not model these from the perspective of interpretability. Specifically, it cannot attribute a perceived importance value to different entities and relationships in the knowledge graph. Thus, the embedded information remains uninterpretable. For example, biological processes, such as gene expression or gene regulation, may have different importance in distinct SL interactions. The clinical utilization of SL prediction based on machine learning is limited by the lack of information regarding its biomedical relevance, its interpretability. Consequently, novel computational approaches are needed to solve this defect (Wang, 2022).

To solve this problem, we propose an improved approach, SLGNN, for SL prediction that models the preferences of genes in

distinct relationships in the knowledge graph, and thus allowing a better understanding of the underlying biological mechanisms. SLGNN is based on an assumption that SL interactions are caused by different molecular events or biological processes, such as gene regulation and co-involvement of genes in biological pathways. We refer to them as SL-related factors that lead to SL interactions. For the convenience of description, ‘factor’ in the following text refers to ‘SL-related factor’. For example, Poly (adenosine diphosphate-ribose) polymerase 1 and breast cancer 1 represent a biologically and medically relevant SL pair, with both molecules being involved in DNA repair. The recognition of the interaction between these molecules in the same biological process led to the development of the first SL-based PARP inhibitor for the treatment of cancer (Lord and Ashworth, 2017). In this example, the DNA repair process appears to be the decisive factor in SL interaction between the two genes, giving an intuitive insight into this SL mechanism. By modeling the preferences of genes for different SL-related factors, the model becomes more interpretable, giving insight into the underlying biology. The computational process underpinning SLGNN is divided into three parts: first, we model the combination of relationships in the gene-related knowledge graph as SL-related factors leading to SL interaction. Second, we obtain the initial embeddings of genes through the explicit message aggregation of the knowledge graph; third, we obtain the final gene embeddings through an SL graph constructed using known SL gene pairs and based on the factor-based message aggregation mechanism. Finally, a supervised end-to-end training model is used for SL interaction prediction. In comparisons the proposed SLGNN outperformed five other state-of-the-art methods, while providing better interpretability.

2 Materials and methods

In this section, we introduce the data used and the formulation of the problem. Then, we describe the details of each module of SLGNN.

2.1 Data description

SynLethDB (Guo *et al.*, 2016) is a database containing a large number of currently known SL gene pairs. It was derived from biochemical analysis, correlation databases and data mining. The latest version of SynLethDB contains a gene-related knowledge graph called SynLethKG. This provides a graphic representation of 36 402 SL interactions between 10 004 genes. Because the SynLethDB only contains positive samples of SL interactions, to alleviate the impact of distribution differences between positive and negative samples, we used the method utilized in KG4SL to generate negative samples with the same number as the positive numbers. SynLethKG is a SL-related biomedical knowledge graph containing 11 entities and 24 relationships related to genes. The details of SynLethKG are shown in Tables 1 and 2.

Table 1. Numbers of the entities in the SynLethKG

Type	No. of entities
Anatomy	400
Biological process	12 703
Cellular component	1670
Compound	2065
Disease	136
Gene	25 260
Molecular function	3203
Pathway	2069
Pharmacologic class	377
Side effect	5702
Symptom	427

Table 2. Numbers of the relationships in the SynLethKG

Type	No. of entities
(Gene, participates, cellular component)	97 652
(Gene, participates, biological process)	619 712
(Anatomy, expresses, gene)	617 175
(Gene, regulates, gene)	267 302
(Gene, interacts, gene)	147 638
(Disease, associates, gene)	24 328
(Gene, participates, molecular function)	110 042
(Gene, covaries, gene)	62 966
(Gene, participates, pathway)	57 441
(Disease, upregulates, gene)	7730
(Compound, causes, side effect)	139 428
(Compound, binds, gene)	16 323
(Anatomy, upregulates, gene)	26
(Disease, presents, symptom)	3401
(Disease, localizes, anatomy)	3373
(Compound, treats, disease)	752
(Disease, resembles, disease)	404
(Disease, downregulates, gene)	7616
(Compound, upregulates, gene)	19 200
(Compound, downregulates, gene)	21 526
(Compound, resembles, compound)	6266
(Pharmacologic class, includes, compound)	1205
(Compound, palliates, disease)	384
(Anatomy, downregulates, gene)	31

2.2 Problem formulation

SL graph. We intended to model SL interaction as a graph $S = (V, E)$, where V is the set of n genes and E is the set of SL interactions. We denote $A \in \mathbb{R}^{n \times n}$ as the adjacency matrix of this graph. If v_i and v_j have a SL interaction, then, A_{ij} is equal to 1, otherwise A_{ij} is equal to 0, modeling interactions between genes as a SL graph.

Knowledge graph. KG stores the attributes of real-world entities and the correlation between these entities in the form of heterogeneous information networks. The correlations are modeled as distinct relationships (Shi *et al.*, 2017). We denote SynLethKG as $G = (V_e, E_r)$, where V_e represents the entities in KG and E_r represents the relationships in KG. KG can be represented as a collection of triples $G = \{(h, t, r) | h, t \in V_e, r \in E_r\}$, where each triple (h, t, r) represents the relationships r between head entity h and tail entity t . It is worth noting that we set the edges in the knowledge graph to be undirected, which means that the number of relationships in the KG is doubled.

Task description. Given SL graph S and SynLethKG G , our task is to automatically generate gene embeddings through SynLethKG G , then perform link prediction in the SL graph S to identify potential SL pairs.

2.3 Overview of SLGNN

The overall framework of the model, which consists of three main components, is illustrated in Figure 1.

Factor modeling. We assume that the SL interactions are caused by different SL-related factors and model these factors to enhance the interpretability of the model.

Knowledge graph message aggregation. Through knowledge graph convolutional networks, we perform explicit message aggregation on entity embeddings in knowledge graphs, the embeddings of genes at the knowledge graph level are obtained.

Factor-based message aggregation. We take the gene embedding obtained from the knowledge graph as the input and perform a message aggregation on the SL graph, incorporating the preference of a given gene for different factors into the aggregation process.

2.3.1 Factor modeling

We aim to obtain SL-related factors that determine SL interaction, representing the commonness of all SL interactions. Although we can simply represent these factors as vectors, it is difficult to assign explicit biological meaning to these. Inspired by KGIN (Wang *et al.*, 2021), we model these factors as combinations of relationships in the KG. This approach derives the embeddings of factors directly from the relationships in KG, providing them with a meaning that can be explained. Suppose P is a factor set shared by all genes. Technically, for each factor $p \in P$, we use the following formula to derive its embedding:

$$e_p = \sum_{r \in E_r} \alpha(r, p) e_r, \quad (1)$$

where e_r is the embedding of a relationship in KG, and $\alpha(r, p)$ is a trainable parameter. In this way, we obtain $|P|$ different factor embeddings.

Different factors should represent independent information (Ma *et al.*, 2019). If the embedding of a factor can be represented by the embeddings of several factors, then this factor is likely to be redundant. On the contrary, if these factors are independent of each other than these they will contain more information. Here, we use distance correlation (Székely *et al.*, 2007) as a regularizer to maintain independence between factor embeddings. Distance correlation measures the correlation between two variables and equals zero if, and only if, the two variables are independent of each other. By minimizing the distance correlation between factor embeddings, the correlation between factors can be reduced, according to the following formula:

$$\mathcal{L}_{\text{IND}} = \sum_{p, p' \in P, p \neq p'} \text{dCor}(e_p, e_{p'}), \quad (2)$$

where $\text{dCor}(\cdot)$ is the distance correlation between two different factor embeddings:

$$\text{dCor}(e_p, e_{p'}) = \frac{\text{dCov}(e_p, e_{p'})}{\sqrt{\text{dVar}(e_p) \cdot \text{dVar}(e_{p'})}}, \quad (3)$$

where $\text{dCov}(\cdot)$ is the distance covariance and $\text{dVar}(\cdot)$ is the distance variance of factor embeddings. By optimizing the above objectives, different factors can generate effective boundaries, improving the interpretability of the model.

2.3.2 Knowledge graph message aggregation

We obtain the initial embedding of a gene through message aggregation of the KG, which avoids the manual design of gene features and can generate features through the rich information represented in the KG. KGNN (Lin *et al.*, 2020) used a knowledge graph convolutional network to automatically generate entity embeddings and we follow this message aggregation mechanism with slight modifications.

Relationships in KG play an important role in the knowledge graph convolution network based on the message passing mechanism, as they connect entities. A notable feature of KG is that the same entities could be linked through different relationships that correspond to distinct biological processes. Therefore, it is necessary to distinguish different relationships during message aggregation. In previous work, KGNN used an attention mechanism to model the KG relationships as decay factors, in order to control the influences of different neighbors. To enhance the performance of the model, we take a different approach to aggregate messages for different relationships in a process that is explicit:

$$e_i^{(1)} = \frac{1}{|\mathcal{N}_i|} \sum_{(r, j) \in \mathcal{N}_i} e_r \odot e_j^{(0)}, \quad (4)$$

where $\mathcal{N}_i = \{(r, j) | (i, r, j) \in G\}$ is the neighbor entity set of entity i , e_r is the embedding of relationship r , $e_j^{(0)}$ is the embedding of entity j

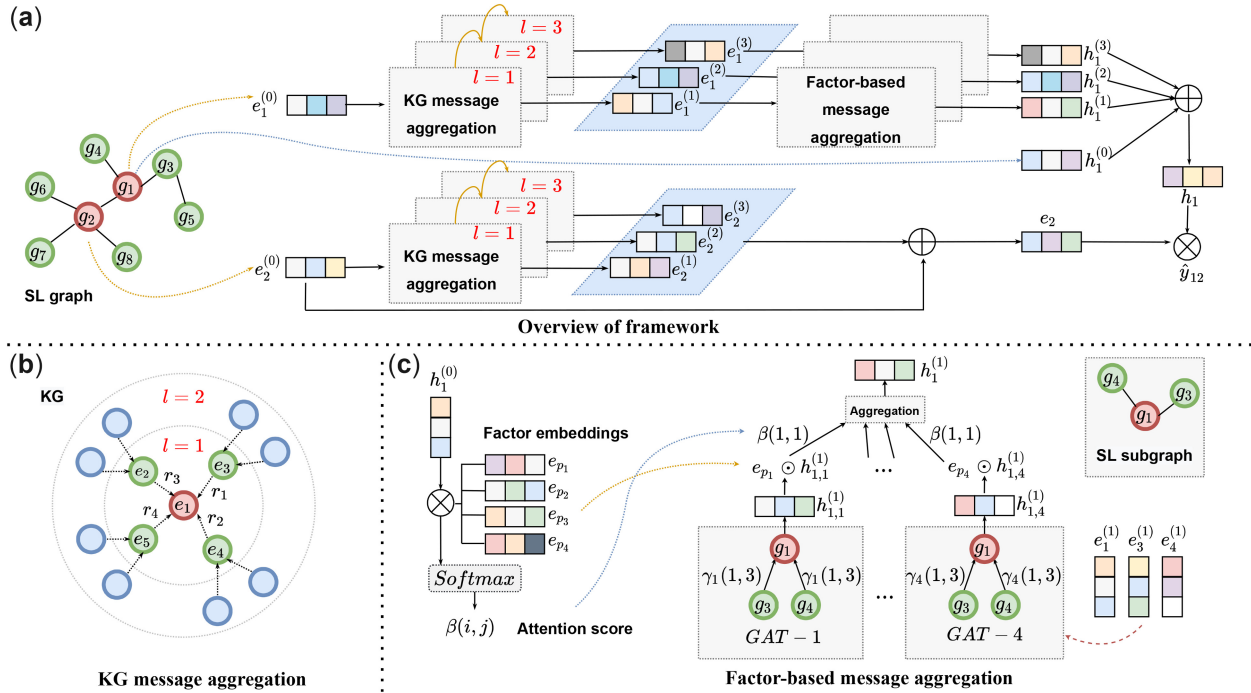


Fig. 1. The overall architecture of SLGNN. (a) Overview of the framework. Here g_i represents a gene node in the SL graph and the initial gene entity embedding $e_i^{(0)}$ is used as input. e_i represents the gene embedding obtained by KG message aggregation and h_i represents the gene embedding obtained by factor-based message aggregation. (b) KG message aggregation. The figure shows the first-order and second-order subgraphs of entity e_1 in the KG, and the neighbors are connected to e_1 through the relationship r_i . (c) Factor-based message aggregation. Gene embedding $h_i^{(0)}$ and factor embeddings e_p receive an attention score β through the attention mechanism. The gene embeddings obtained by KG message aggregation $e_i^{(l)}$ are input into four different GATs to get embeddings $h_{i,p}^{(l)}$, where γ represents the attention score in the GAT. Finally, the gene embedding $h_i^{(l)}$ is obtained by aggregation

and \odot is an element-wise product. This way, first-order neighbor messages for each entity can be aggregated. To aggregate the messages of the higher-order neighbors, we recursively set the knowledge graph convolution network to a multilayer network:

$$e_i^{(l+1)} = \frac{1}{|\mathcal{N}_i^l|} \sum_{(r,j) \in \mathcal{N}_i^l} e_r \odot e_j^{(l)}. \quad (5)$$

Through this l -layer convolution network, the embeddings of entities in KG are obtained and we focus on the derived gene embedding by the sum of the representation of each layer:

$$e_i = e_i^{(0)} + e_i^{(1)} + \dots + e_i^{(L)}, \quad (6)$$

where L is the number of layers.

2.3.3 Factor-based message aggregation

After obtaining the initial gene embedding through KG message aggregation, we use a factor-based message aggregation to derive gene embeddings in the SL graph. Specifically, gene embedding $e_i^{(l+1)}$ of the $l+1$ layer of the KG convolutional network is used to obtain the corresponding gene embedding $h_i^{(l+1)}$.

For a given a gene, individual factors make a different contribution to the SL interaction associated with it. We use an attention mechanism to measure the importance of these factors for a given gene. For gene $i \in V$, the attention score $\beta(i, p)$ is:

$$\beta(i, p) = \frac{\exp(e_p^T h_i^{(l)})}{\sum_{p' \in P} \exp(e_{p'}^T h_i^{(l)})}, \quad (7)$$

where e_p is the embedding of factor p and $h_i^{(l)}$ is the embedding of the l -th layer obtained by gene i after the factor-based message aggregation, while $h_i^{(0)}$ is a randomly initialized embedding.

Intuitively, gene i should generate different embeddings for each factor before message aggregation. Therefore, we use graph

attention networks (GATs) (Veličković et al., 2017) to aggregate messages in the SL graph, adaptively deriving different gene embeddings. We input the gene embeddings of layer $l+1$ of the knowledge graph convolutional network into $|P|$ different GATs, deriving the $h_{i,p}^{(l+1)}$ corresponding to the layer $l+1$:

$$h_{i,p}^{(l+1)} = \sum_{j \in \mathcal{N}_i^l} \gamma_p(i, j) e_j^{(l+1)}, \quad (8)$$

$$\gamma_p(i, j) = \frac{\exp(\sigma(a_p^T [W_p e_i^{(l+1)} \| W_p e_j^{(l+1)}]))}{\sum_{k \in \mathcal{N}_i^l} \exp(\sigma(a_p^T [W_p e_i^{(l+1)} \| W_p e_k^{(l+1)}]))}, \quad (9)$$

where $\gamma_p(i, j)$ is the attention score in GAT- p , $\mathcal{N}_i^l = \{j | (i, j) \in S\}$ is the neighbor set of gene i in SL graph, $e_j^{(l+1)}$ is the embedding of gene j obtained by $l+1$ layer of knowledge graph convolution network, W_p is a projection matrix, a_p is a weight vector that is different for each factor and σ is an activation function, here, we use LeakyReLU. Finally, we further aggregate these gene embeddings generated by the GATs based on the attention scores of factors:

$$h_i^{(l+1)} = \sum_{p \in P} \beta(i, p) e_p \odot h_{i,p}^{(l+1)}, \quad (10)$$

where $h_{i,p}^{(l+1)}$ is the embedding of gene i obtained by GAT- p , e_p is the embedding of factor p , \odot is the element-wise product and $\beta(i, p)$ is the attention score of gene i to factor p . Following the same strategy used for factor modeling, we attempt to make different GAT-generated embeddings independent of each other to express more information:

$$\mathcal{L}_{\text{GAT}} = \sum_{p, p' \in P, p \neq p'} \text{dCor}(h_p^{\text{avg}}, h_{p'}^{\text{avg}}), \quad (11)$$

where h_p^{avg} represents the average embedding of all genes obtained by GAT- p . Finally, we add each layer of gene embedding to obtain the final gene embedding h_i :

$$\mathbf{b}_i = \mathbf{b}_i^{(0)} + \mathbf{b}_i^{(1)} + \dots + \mathbf{b}_i^{(L)}, \quad (12)$$

where L is the number of layers.

2.4 Model optimization

The inner product of the two embeddings \mathbf{b}_i and \mathbf{e}_j for the gene pair i, j is used as the probability of a SL interaction. The cross-entropy is calculated using the probability value \hat{y}_{ij} and the truth label:

$$\hat{y}_{ij} = \sigma((\mathbf{b}_i)^T \mathbf{e}_j), \quad (13)$$

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{E}_t|} \sum_{i,j \in \mathcal{E}_t} y_{ij} \cdot \ln(\hat{y}_{ij}) + (1 - y_{ij}) \cdot \ln(1 - \hat{y}_{ij}), \quad (14)$$

where σ is sigmoid function. \mathcal{E}_t is the set of SL interactions in the training set.

Independence losses \mathcal{L}_{IND} , \mathcal{L}_{GAT} and $L2$ -normalization losses of the model parameter $\mathcal{L}_{\text{norm}}$ must also be considered. Regularization parameters include the embeddings of entities, the embeddings of relationships and the projection matrixes. The final loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda_1(\mathcal{L}_{\text{IND}} + \mathcal{L}_{\text{GAT}}) + \lambda_2 \mathcal{L}_{\text{norm}}, \quad (15)$$

where λ_1 and λ_2 are two hyperparameters that control the independence loss and the $L2$ -normalization loss, respectively.

3 Results

In this section, we first introduce the state-of-the-art baseline methods and the implementation details, followed by parameter sensitivity analysis and model ablation study. Finally, we give an example of model interpretability.

3.1 Experimental setups

3.1.1 Baseline methods

To validate the performance of SLGNN, we compare it to some recently published state-of-the-art SL interaction prediction methods. The first and second of these are based on MF, the third and fourth on graph convolution network, while the final one represents a knowledge graph convolution network. Note that, the first four methods do not use KGs to generate gene embeddings:

- SL^2MF (Liu *et al.*, 2020) predicts SL interactions based on logical MF.
- CMF (Liany *et al.*, 2020) uses CMF to model multiple matrices.
- DDGCN (Cai *et al.*, 2020) predicts sparse SL interaction based on dual-dropout GCN.
- GCATSL (Long *et al.*, 2021) adopts a context-based attention network for SL prediction.
- KG4SL (Wang *et al.*, 2021) represents the first KG enhanced SL interaction prediction model.

3.1.2 Implementation details

The dataset was randomly divided into training set, validation set and test set at a ratio of 8:1:1. To make the experimental result more convincing, we used 5-fold cross-validation for all methods. For all baselines, we used the hyperparameters and model parameters described in the original paper. Our proposed SLGNN was implemented in Python3.9 and PyTorch1.10. The GATs were implemented using the DGL (Wang *et al.*, 2019) framework, and parameters, such as dropout rate and negative slope, were set according to default settings in the original GAT paper. The model learning rate was set to 0.002 and an early stopping strategy was used. Due to memory constraints, we randomly dropout edges in the KG with a ratio of 0.5. In the KG, we used word embeddings in PyTorch to obtain unique embeddings for entities and

relationships, and a random initialization applying a standard normal distribution.

A greedy strategy was used to confirm the optimal hyperparameters. At first, initial hyperparameters were set from related experience, then, each hyperparameter was optimized one by one. Specifically, the coefficients of constraints λ_1 and λ_2 were tuned in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, the number of GNN layers was tuned in $\{1, 2, 3\}$, the number of factors was tuned in $\{1, 2, 4, 8\}$ and the dimension size was tuned in $\{32, 64, 128, 256\}$.

3.2 Performance evaluation

We compare the performance of SLGNN with the five previously described baseline models according to three evaluation metrics: AUC, AUPR and $F1$ -score. As shown in Table 3, SLGNN outperforms all baseline methods. The AUC, AUPR and $F1$ -score of SLGNN is 0.9635, 0.9710 and 0.9089, respectively, i.e. 2.1%, 1.8% and 2.9% higher than the second-best model. Compared to MF methods with shallow embedding, GNN-based methods take advantage of the topological structure of the graph to propagate information, and exploit the similarity of known SL interactions, thus improving model performance. The comparisons indicate that the two KG-based methods, KG4SL and our proposed SLGNN, exhibit greatly improved performance as they benefit from the enhanced embedding facilitated by the rich semantic knowledge present in the KG. However, to generate gene embeddings, KG4SL only uses an attention mechanism to distinguish different relationships in gene sub-graphs, without considering the semantic information of relationship embeddings. From the comparisons carried out it is clear that SLGNN circumvents this limitation and performs better at generating gene embeddings for the purposes of SL interaction prediction.

3.3 Parameter sensitivity

In this section, we explore the influence of some parameters on the performance of the model. Though experiments, SLGNN achieves the best results when the number of GNN layers is 3, the number of factors is 4, the weight factor λ_1 is 10^{-4} , λ_2 is 10^{-3} and the embedding dimension is 256. Therefore, we use these optimal hyperparameters as the default setting while performing sensitivity analyses on each hyperparameter. As illustrated in Figure 2, the performance of the model increases with an increasing dimension setting. A higher number of factors initially improve performance that reach its peak at four. Increasing the number further results in a slow decline in the performance. Intuitively, as the number of network layers increases, the gene entities in the knowledge graph can perceive entity information that is farther away, helping to improve model performance. The result shown in Figure 2 is consistent with this intuition. Regarding the $L2$ -normalization parameter λ_2 , it can be seen from the figure that the model performance initially improved with increasing weight factor, reach its optimum, and then start to decline.

We also explore the influence of the hyperparameter λ_1 controlling independence loss. As shown in Table 4, when λ_1 increases, the distance correlation between factor embeddings also grows. This is also accompanied by a slight decrease of the AUC, indicating that

Table 3. Comparing the performance of the different methods (the best result is shown in bold and the second-best result is underlined)

Methods	AUC	AUPR	$F1$
SL^2MF	0.7912 ± 0.0024	0.8637 ± 0.0054	0.7478 ± 0.0077
CMF	0.8023 ± 0.0035	0.8422 ± 0.0042	0.7565 ± 0.0016
DDGCN	0.8413 ± 0.0089	0.8851 ± 0.0065	0.8133 ± 0.0065
GCATSL	0.9214 ± 0.0022	0.9450 ± 0.0033	0.8631 ± 0.0076
KG4SL	<u>0.9436 ± 0.0019</u>	<u>0.9539 ± 0.0008</u>	<u>0.8833 ± 0.0017</u>
SLGNN	0.9635 ± 0.0017	0.9710 ± 0.0010	0.9089 ± 0.0017

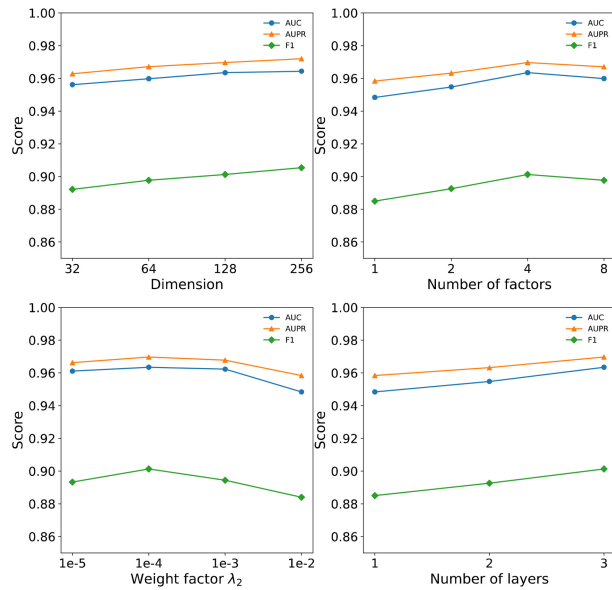


Fig. 2. The effect of various hyperparameters of the model. This figure shows the impact of four hyperparameters on model performance, including the embedding dimension, the number of factors, the number of layers and the weight factor λ_2 controlling L_2 -normalization

Table 4. The impact of independence between the factors

λ_1	AUC	AUPR	Distance correlation
10^{-5}	0.9598 ± 0.0024	0.9656 ± 0.0016	2.3653
10^{-4}	0.9612 ± 0.0035	0.9688 ± 0.0019	1.0275
10^{-3}	0.9635 ± 0.0017	0.9710 ± 0.0010	0.0245
10^{-2}	0.9622 ± 0.0013	0.9697 ± 0.0012	0.0043

Table 5. Comparison of SLGNN and model variants

Model	AUC	AUPR	F1
SLGNN	0.9635 ± 0.0017	0.9710 ± 0.0010	0.9089 ± 0.0017
SLGNN _{w/o F}	0.9345 ± 0.0012	0.9422 ± 0.0041	0.8715 ± 0.0032
SLGNN _{w/o G}	0.9558 ± 0.0021	0.9664 ± 0.0013	0.8915 ± 0.0016

when the factors are closer to each other they contained more semantic information, improving model performance.

3.4 Ablation study

In this section, we explore the relative contribution of individual components to the performance of the model, by eliminating individual components one by one.

First, we verify the influence of factor modeling. To do so, we create a variant of the model, SLGNN_{w/o F} in which the factor modeling component is removed resulting in the direct use of GAT to aggregate messages in the SL graph. When the SLGNN_{w/o F} is compared to the original model, it results a significant reduction in model performance. As shown in Table 5, the altered model is inferior to KG4SL, clearly indicating the contribution of factor modeling.

Next, we explore the importance of GATs to the performance of the model. Similarly, we create another variant of the model SLGNN_{w/o G}, which does not use GATs in the factor-based message aggregation process. As shown in Table 5, the performance of this variant to the model is also inferior, clearly indicating that our strategy of using GAT to generate different gene embeddings with preserved independence is an effective way to get a better performance.

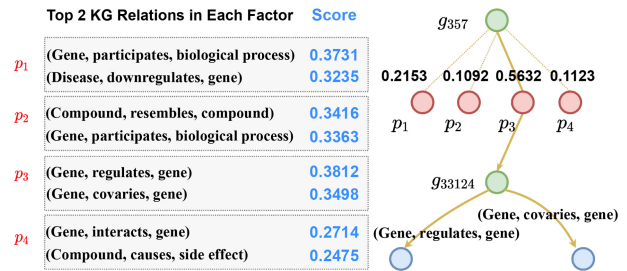


Fig. 3. An example of SGNN interpretability. The top-2 KG relationships and their weights in the factors are shown on the left, an example of SL interactions between gene pairs indicated by factors is shown on the right

3.5 Interpretability of SLGNN

In this section, we explain the semantics of factors used in the model and give an example to illustrate the interpretability of result obtained by SLGNN.

It is readily apparent that not all factors influence an interaction that results in SL. Thus, various combinations of factors in the model are weighted differently to reflect the importance of distinct factors. Of the possible relationships illustrated in Figure 3, in the composition of factor p_1 two particular relationships (gene, participates, biological process) and (disease, downregulates, gene) carry the highest weights. This suggests that relationships describing the interaction between genes and disease and the participation of genes in biological processes are the most likely to contain a factor leading to SL. As can be seen from the figure, by maintaining the independence between the factors, the KG relationship with the highest weight is different in each factor, improving the interpretability of the model.

We can explain the interpretability of our proposed model from another perspective. The importance of factors leading to SL interactions differs for each gene. In other words, genes have certain bias to given factors. As can be seen from the figure, $gene_{357}$ and $gene_{33124}$ result in a SL interaction that can be derived through the attention mechanism. In this example, factor p_3 has the highest attention score indicating that p_3 is the main factor that causes the SL interaction between $gene_{357}$ and $gene_{33124}$. From the semantics of factor p_3 , we can explore the specific mechanism of SL interaction between these two genes.

4 Conclusion and discussion

In this article, we propose a knowledge graph-based model SLGNN for the prediction of SL interactions, while also modeling the SL-related factors that lead to SL interactions to improve the biological interpretability of the results. First, the factors leading to SL interaction are modeled as weighted sums of different relationships in KG, and the independence of factors is maintained by minimizing the distance correlation between them. Then, the GNN-based message aggregation mechanism is used to obtain initial embeddings for the genes in KG. Finally, we conduct a factor-based message aggregation through the SL graph and use GATs to improve the performance of the model. The experimental results show that our proposed SLGNN is superior to five existing state-of-the-art methods in predicting SL interactions.

Despite its remarkable performance, SLGNN still has some limitations. For example, gene embeddings obtained through the KG graph is coupled and this has an impact on both interpretability and performance. To further improve this drawback in future work, we will attempt to use the decoupling representation learning, similar to that described in DisenKGAT (Wu et al., 2021), to achieve higher quality gene embeddings. In addition, due to the sparsity of the SL graph, negative samples were created using theoretical considerations and may contain potential positive samples. In this work, we simply generated negative samples by randomly sampling. This strategy is essentially unreliable and its use in the training of the model may reduce the accuracy of predictions. Adopting alternative

strategies, such as contrastive learning, may alleviate the problem of the imbalance between positive and negative samples, improving quality of negative sampling (Jiang *et al.*, 2021). Obtaining reliable negative samples is one of our future research goals.

Funding

This work was supported by the grants from the National Key R&D Program of China [2021YFA0910700]; Shenzhen Science and Technology University Stable Support Program [GXWD20201230155427003-20200821222112001]; Shenzhen Science and Technology Program [JCYJ20200109113201726]; Guangdong Basic and Applied Basic Research Foundation [2021A1515012461, 2021A1515220115]; and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies [2022B1212010005].

Conflict of Interest: none declared.

References

- Benstead-Hume, G. *et al.* (2019) Predicting synthetic lethal interactions using conserved patterns in protein interaction networks. *PLoS Comput. Biol.*, **15**, e1006888.
- Cai, R. *et al.* (2020) Dual-dropout graph convolutional network for predicting synthetic lethality in human cancers. *Bioinformatics*, **36**, 4458–4465.
- Das, S. *et al.* (2019) DiscoverSL: an R package for multi-omic data driven prediction of synthetic lethality in cancers. *Bioinformatics*, **35**, 701–702.
- Du, D. *et al.* (2017) Genetic interaction mapping in mammalian cells using CRISPR interference. *Nat. Methods*, **14**, 577–580.
- Guo, J. *et al.* (2016) SynLethDB: synthetic lethality database toward discovery of selective and sensitive anticancer drug targets. *Nucleic Acids Res.*, **44**, D1011–D1017.
- Hamilton, W.L. (2020) *Graph Representation Learning*. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Vol. 14, Springer, Cham. pp. 1–159.
- Huang, J. *et al.* (2019) Predicting synthetic lethal interactions in human cancers using graph regularized self-representative matrix factorization. *BMC Bioinformatics*, **20**, 1–8.
- Jerby-Arnon, L. *et al.* (2014) Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality. *Cell*, **158**, 1199–1209.
- Jiang, Z. *et al.* (2021) Improving contrastive learning on imbalanced data via open-world sampling. In: *Advances in Neural Information Processing Systems*, Virtual Conference, December 6–14, 2021. Vol. 34, pp. 5997–6009.
- Liany, H. *et al.* (2020) Predicting synthetic lethal interactions using heterogeneous data sources. *Bioinformatics*, **36**, 2209–2216.
- Lin, X. *et al.* (2020) KGNN: knowledge graph neural network for drug-drug interaction prediction. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, Yokohama, Japan*. pp. 2739–2745.
- Liu, Y. *et al.* (2020) SL²MF: predicting synthetic lethality in human cancers via logistic matrix factorization. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **17**, 748–757.
- Long, Y. *et al.* (2021) Graph contextualized attention network for predicting synthetic lethality in human cancers. *Bioinformatics*, **37**, 2432–2440.
- Lord, C.J. and Ashworth, A. (2017) PARP inhibitors: synthetic lethality in the clinic. *Science*, **355**, 1152–1158.
- Luo, J. *et al.* (2009) A genome-wide rai screen identifies multiple synthetic lethal interactions with the ras oncogene. *Cell*, **137**, 835–848. <https://doi.org/10.1016/j.cell.2009.05.006>.
- Ma, J. *et al.* (2019) Disentangled graph convolutional networks. In: *Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, USA*. pp. 4212–4221. Long Beach, California, USA.
- Paladugu, S.R. *et al.* (2008) Mining protein networks for synthetic genetic interactions. *BMC Bioinformatics*, **9**, 1–14.
- Shi, C. *et al.* (2017) A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.*, **29**, 17–37.
- Sinha, S. *et al.* (2017) Systematic discovery of mutation-specific synthetic lethals by mining pan-cancer human primary tumor data. *Nat. Commun.*, **8**, 1–13.
- Székely, G.J. *et al.* (2007) Measuring and testing dependence by correlation of distances. *Ann. Stat.*, **35**, 2769–2794.
- Topatana, W. *et al.* (2020) Advances in synthetic lethality for cancer therapy: cellular mechanism and clinical translation. *J. Hematol. Oncol.*, **13**, 1–22.
- Veličković, P. *et al.* (2018) Graph attention networks. In: *Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada*.
- Wang, J. *et al.* (2022) Computational methods, databases and tools for synthetic lethality prediction. *Brief. Bioinform.*, **23**, bbac106.
- Wang, M. *et al.* (2019) Deep graph library: a graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315.
- Wang, S. *et al.* (2021) KG4SL: knowledge graph neural network for synthetic lethality prediction in human cancers. *Bioinformatics*, **37**, i418–i425.
- Wang, X. *et al.* (2021) Learning intents behind interactions with knowledge graph for recommendation. In: *Proceedings of the Web Conference 2021, New York, NY, USA*. pp. 878–887.
- Wu, J. *et al.* (2021) DisenKGAT: knowledge graph embedding with disentangled graph attention network. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Gold Coast, Queensland, Australia*, pp. 2140–2149.
- Zhang, B. *et al.* (2021) The tumor therapy landscape of synthetic lethality. *Nat. Commun.*, **12**, 1–11.