Bioinformatics, 37(18), 2021, 2988–2995
doi: 10.1093/bioinformatics/btab207
Advance Access Publication Date: 26 March 2021
Original Paper

Data and text mining

SumGNN: multi-typed drug interaction prediction via efficient knowledge graph summarization

Yue Yu[1,†], Kexin Huang [2,†], Chao Zhang[1], Lucas M. Glass[3,4], Jimeng Sun[5] and Cao Xiao[3,*]

[1]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA, [2]Health Data Science, Harvard T.H. Chan School of Public Health, Boston, MA 02115, USA, [3]Analytic Center of Excellence, IQVIA, Cambridge, MA 02139, USA, [4]Department of Statistics, Temple University, Philadelphia, PA 19122, USA and [5]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: Jonathan Wren

Received on October 22, 2020; revised on February 7, 2021; editorial decision on March 17, 2021; accepted on March 24, 2021

Abstract

Motivation: Thanks to the increasing availability of drug–drug interactions (DDI) datasets and large biomedical knowledge graphs (KGs), accurate detection of adverse DDI using machine learning models becomes possible. However, it remains largely an open problem how to effectively utilize large and noisy biomedical KG for DDI detection. Due to its sheer size and amount of noise in KGs, it is often less beneficial to directly integrate KGs with other smaller but higher quality data (e.g. experimental data). Most of existing approaches ignore KGs altogether. Some tries to directly integrate KGs with other data via graph neural networks with limited success. Furthermore most previous works focus on binary DDI prediction whereas the multi-typed DDI pharmacological effect prediction is more meaningful but harder task.

Results: To fill the gaps, we propose a new method SumGNN: *knowledge summarization graph neural network*, which is enabled by a subgraph extraction module that can efficiently anchor on relevant subgraphs from a KG, a self-attention based subgraph summarization scheme to generate reasoning path within the subgraph, and a multi-channel knowledge and data integration module that utilizes massive external biomedical knowledge for significantly improved multi-typed DDI predictions. SumGNN outperforms the best baseline by up to 5.54%, and performance gain is particularly significant in low data relation types. In addition, SumGNN provides interpretable prediction via the generated reasoning paths for each prediction.

Availability and implementation: The code is available in Supplementary Material.

Contact: cao.xiao@iqvia.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Adverse drug–drug interactions (DDI) are modifications of the effect of a drug when administered with another drug, which is a common and dangerous scenario for patients with complicated conditions. Undetected adverse DDIs have become serious health threats and caused nearly 74 000 emergency room visits and 195 000 hospitalizations each year in the United States alone (Percha and Altman, 2013). To mitigate these risks and costs, accurate prediction of DDIs becomes a clinically important task. Two types of data are being utilized for developing DDI detection models: Manually curated DDI networks and large biomedical knowledge graphs (KGs).

Curated DDI networks: Researchers have curated DDI networks based on experimental datasets and literature such as TWOSIDES (Tatonetti *et al.*, 2012), MINER (Zitnik *et al.*, 2018a) and DrugBank (Ryu *et al.*, 2018; Wishart *et al.*, 2018). These curated data are *of higher quality but expensive to create and usually smaller in size*.

Knowledge graph: Over the years, large KG such as (Rotmensch *et al.*, 2017), Hetionet (Himmelstein and Baranzini, 2015) and DRKG (Ioannidis *et al.*, 2020) have been constructed from literature mining and database integration. However, these KGs are *large and noisy*: out of their tens of thousands of nodes with millions of edges, only a small subgraph is relevant to a prediction target.

Deep learning: Graph neural networks (GNN) have achieved great performance by casting DDI prediction as a link prediction problem on DDI graphs (Gysi *et al.*, 2020; Huang *et al.*, 2020b; Zitnik *et al.*, 2018b). However, existing deep learning models are often trained only based on the DDI dataset at hand, ignoring the large biomedical KG (Himmelstein and Baranzini, 2015; Ioannidis *et al.*, 2020) which can benefit the DDI predictions since DDI is driven by complicated biomedical mechanism. Some recent works (Karim *et al.*, 2019; Lin *et al.*, 2020) tried to integrate KG into the DDI prediction via direct integration of standard KG and GNN methods. But DDI prediction presents unique modeling difficulties since the input KG is large and noisy while the pertinent information for a drug pair is local. Moreover, most existing works also only make binary classification—predicting the presence of DDIs, despite that predicting the particular DDI type is a more meaningful task.

Our approach. In this work, we propose a new method SumGNN that efficiently uses KG to aid drug interaction prediction. SumGNN enjoys improved predictive performance, efficiency, inductiveness and interpretability. SumGNN provides the following technical contributions:

1. **Local subgraph for identifying useful information**. We use local subgraph in the KG around drug pairs to extract useful information, instead of the entire KG. The subgraph formulation allows noise reduction by anchoring on relevant information and is highly scalable since the message passing receptive field is significantly decreased.
2. **Subgraph summarization scheme for generating reasoning path.** We then propose a summarization scheme to generate mechanism pathway for drug interactions. We develop a layer-independent self-attention mechanism to generate signal intensity score for each edge in the subgraph and prune out a KG subgraph pathways that have high scores. As this pruned subgraph is sparse, it provides insights on the biological processes that drive drug interactions.
3. **Multi-channel data and knowledge integration for improved multi-typed DDI predictions**. We propose to use multi-channel neural encoding to aggregate diverse set of data sources, ranging from the summarized subgraph embedding to chemical structures. It enables utilization of massive external biomedical knowledge for significantly improved multi-typed DDI predictions. In addition, the neural encoding takes different subgraph in each propagation, forming an inductive bias that promotes generalizability in low-resource DDI types.

We conduct extensive experiments to show SumGNN improves DDI prediction significantly. It has up to 5.54% increase over the best baseline in F1 while the inference time is greatly reduced. Moreover, SumGNN excels at low-resource settings whereas previous works do not. SumGNN is also able to provide reasonable clues about the underlying mechanism of the drug interactions.

2 Related works

External knowledge graph integration. Recently, several efforts have attempted to leverage the KG for downstream tasks such as recommendation (Sun *et al.*, 2018; Wang *et al.*, 2019a,b), information extraction (Liang *et al.*, 2020; Wang *et al.*, 2018) and drug interaction prediction (Celebi *et al.*, 2019; Karim *et al.*, 2019; Lin *et al.*, 2020). For drug interaction prediction, Takeda *et al.* (2017) integrate the pharmacokinetic (PK) or pharmacodynamic (PD) side-effect when predicting drug interaction and Li *et al.* (2015) develop a Bayesian network to combine molecular similarity and drug side-effect similarity to predict the drug effect. However, these methods only consider side effect as the external knowledge, which may not be comprehensive enough in our task. With the emergence of the biomedical KGs (Himmelstein and Baranzini, 2015; Ioannidis *et al.*, 2020; Rotmensch *et al.*, 2017), more types of entities and relations have been applied to this task, as Wang (2017), Burkhardt *et al.* (2019), Celebi *et al.* (2019), Karim *et al.* (2019) and Dai *et al.* (2020) project each entity and relation to a dense vector with KG embedding techniques (Bordes *et al.*, 2013; Su *et al.*, 2020; Trouillon *et al.*, 2016) and then feed them to neural networks for prediction. However, they do not directly harness the neighborhood information for target entities during inference, thus the external knowledge information are not sufficiently exploited. To tackle the above drawback (Lin *et al.*, 2020) adopts graph convolutional networks with neighborhood sampling to explicitly model the neighborhood relations with higher inference speed. However, as each neighboring entity could play a crucial role in the drug interaction mechanism, random sampling could potentially dropout these important factors and hinders the prediction performance. In contrast, SumGNN provides a learnable way to extract useful information in the neighborhood. In addition, the previous works all focus on binary DDI prediction whereas SumGNN evaluates on multi-type relation network.

Subgraph graph neural network. Graph neural networks have been proposed for modeling the relation between nodes (Kipf and Welling, 2017; Schlichtkrull *et al.*, 2018; Srinivasa *et al.*, 2020; Veličković *et al.*, 2018; Yu *et al.*, 2020a) and have been successfully applied to various domains (Shang *et al.*, 2019; Shi *et al.*, 2020; Yu *et al.*, 2020b). Subgraph structure contains rich information for many graph learning tasks (Huang and Zitnik, 2020; Teru and Hamilton, 2020; Veličković *et al.*, 2019). For instance, Ego-CNN applies local ego network to identify structures for graph classification (Tzeng and Wu, 2019). Alsentzer *et al.* (2020) formulate a multi-channel way to subgraph classification. Cluster-GCN (Chiang *et al.*, 2019) and GraphSAINT (Zeng *et al.*, 2020) use subgraphs to improve GNN scalability. More relevant to us, Zhang and Chen (2018) apply local subgraph for link prediction and GraIL (Teru and Hamilton, 2020) extend this idea into KG completion task via utilizing multi-relational information. In contrast, SumGNN is driven by the domain DDI prediction problem and is the first to design a graph summarization module on subgraphs to obtain tractable pathway. It also integrates a new multi-channel neural encoding mechanism. We show these modules significantly improve predictive performance over these related works in Section 4.2.

3 Materials and methods

We present SumGNN in this section (The SumGNN code is available at https://github.com/yueyu1030/SumGNN.). We summarize problem settings in Section 3.1 and describe our method in detail in Section 3.2. Our method can be decomposed into three modules. First, we extract the local subgraph in the KG around drug pairs to obtain useful information. Then, we propose a summarization scheme to generate a mechanism pathway for drug interactions. After that, we describe a multi-channel neural encoding layer to predict the pharmacological effect.

3.1 Problem settings

Definition 1(**Drug Interaction Graph**). Given drugs $\mathcal{D}$ and pharmacological effects $\mathcal{R}_D$, the drug interaction graph $\mathcal{G}_{DDI}$ is defined as a set of triplets $\mathcal{G}_{DDI} = \{(u, r, v)|u \in \mathcal{D}, r \in \mathcal{R}_D, v \in \mathcal{D})\}$, where each triplet ($u$, r, v) represents that drug u and drug v have pharmacological effect r.

Definition 2(**External Biomedical Knowledge Graph**). Given a set of various biomedical entities $\mathcal{E}$ and the biomedical relation among the entities $\mathcal{R}$, the external biomedical KG $\mathcal{G}_{KG}$ is defined as $\mathcal{G}_{KG} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$ with each item (h, r, t) describes a biomedical relation r between entity h and entity t. Note that we aggregate the drug entities in $\mathcal{G}_{DDI}$ to $\mathcal{G}_{KG}$, i.e. $\mathcal{R}_D \in \mathcal{R}$, and $\mathcal{D} \in \mathcal{E}$.

Problem 1(**Multi-relational DDI Prediction**). The Drug–drug interaction (DDI) prediction is to output the pharmacological effect given the a

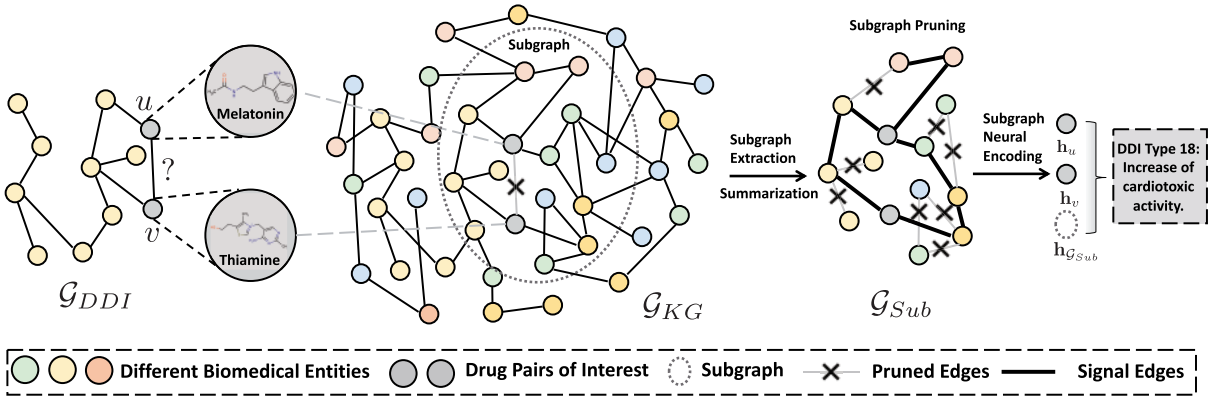

Fig. 1. SumGNN illustration

pair of drugs. Mathematically, it is to learn a mapping $\mathcal{F}: \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{R}_D$ from a drug pair $(u, v) \in (\mathcal{D} \times \mathcal{D})$ to the pharmacological effect $r \in \mathcal{R}_D$.

3.2 The SumGNN method

SumGNN is composed of three modules: subgraph anchoring, knowledge summarization and multi-channel neural encoding. For a given drug pair, we anchor to a subgraph of potential biomedical entities that are close to the pairs in the KG. Then, we propose a new graph neural network that has a summarization scheme to provide a condense pathway to reason about the drug interaction mechanism. Given this pathway graph, we use multi-channel neural encoding, to integrate diverse sources of available information to generate a sufficient drug pair representation. At last, a decoding classifier is followed to predict the interaction outcome. We initialize all entity embedding using KG method TransE (Bordes *et al.*, 2013), where a entity is denoted as $\mathbf{h}_u^{(0)}$ (Fig. 1).

(A) The local subgraph extraction module

The biomedical KG describes the complicated mechanism of human biology. Modulation in several nodes (drug-pairs) in the KG can perturb the connected nodes (e.g. disease, cellular component, etc.) which creates a ripple effect that eventually result in various physiological outcomes (Himmelstein and Baranzini, 2015). The effect is diffused as distance between the drug pairs and the biomedical entities increases. Thus, to understand the drug interactions, we focus on local subgraphs in the KG around the drug pairs. Specifically, for drug pairs u and v, we first extract the k-hop neighboring nodes for both u and v, $\mathcal{N}_k(u) = \{s|d(s,u) \leq k\}$ and $\mathcal{N}_k(v) = \{s|d(s,v) \leq k\}$, where $d(\cdot,\cdot)$ stands for the distance between two nodes on $\mathcal{G}_{KG}$. Then, we obtain the enclosing subgraph based on the intersection of these nodes, $\mathcal{G}_{Sub} = \{(u,r,v) | u,v \in \mathcal{N}_k(u) \cap \mathcal{N}_k(v), r \in \mathcal{R}\}$.

Motivated by Zhang and Chen (2018) which highlights the importance of node relative position to the central node u, v in the subgraph, we augment the initial node embedding in the subgraph by concatenating a position vector. For each node i in the subgraph $\mathcal{G}_{Sub}$, we first compute the shortest path length $d(i, u)$ and $d(i, v)$ between i and the center drug pairs nodes u, v. After that, we convert it into a position vector $\mathbf{p}_i = [\text{one} - \text{hot}(d(i,u)) \oplus \text{one} - \text{hot}(d(i,v))]$. Then, we update the node i representation as $\mathbf{h}_i^{(0)} = [\mathbf{h}_i^{(0)}, \mathbf{p}_i]$.

(B) The knowledge summarization module

To provide biological insights in addition to the predictive outcome, we design a knowledge summarization module to summarize the subgraph information into a graph-based pathway for potential drug interactions. Note that the pathway is not a linear line, but a sparse subgraph since drug interactions are usually due to complicated interplays among many types of biomedical entities. The summarization requires that we need to retain edges that contain most useful signals for the drug interactions while removing paths that are not important. To achieve this, we adopt a *layer-independent*, *relation-aware* self-attention module to assign a weight for every edge in $\mathcal{G}_{Sub}$. These weights are generated based on the input featurization $\mathbf{h}^{(0)}$ and represent the interaction signal intensities for edge pruning.

Specifically, we denote the interaction signal intensity score for the edge connecting any biomedical entity i and j as $\alpha_{i,j}$. Inspired by the relation-aware transformer architecture (Shaw *et al.*, 2018), we use self-attention mechanism, which takes account into all neighbor nodes in the subgraph to generate the attention weight. This attention mechanism is ideal for us because it generates the signal intensity score after examining all biomedical entities in the subgraph around the drug-pairs. Here, $\alpha_{u,v}$ is calculated as

$$\alpha_{i,j} = \text{Threshold}\left(\phi\left(\frac{\mathbf{h}_j^{(0)}\mathbf{W}^J(\mathbf{h}_i^{(0)}\mathbf{W}^I + \mathbf{r}_{ij})^T}{\sqrt{d_k}}\right), \gamma\right), \quad (1)$$

where the $\mathbf{W}^J$ and $\mathbf{W}^I$ are the self-attention key weights that contain representation for each node in the subgraph, r_{ij} encodes the relationship between the two entities i and j, $\sqrt{d_k}$ is the size of feature vector $\mathbf{h}^{(0)}$ for normalization, γ is the signal threshold and $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is the tanh function for non-linear transformation. Intuitively, this function first computes the dot product between $\mathbf{h}_i^{(0)}$ and $\mathbf{W}^I$ to get an attention score between node i and every neighbor node in the subgraph. Then we sum it up with the relation embedding, followed up the same procedure to calculate attention score between node j and every other node through dot product between $\mathbf{h}_j^{(0)}$ and $\mathbf{W}^J$. By taking the dot product with every other nodes for both i, j, the final score considers all the subgraph information. Then, after non-linear transformation, we calculates the signal intensity score ranging from –1 to 1 for this edge. At last, we apply a threshold function to screen out edges that are below an intensity score threshold γ by setting them with weight 0 since they are not important for the interaction prediction and setting them 0 would prune these edges from message passing process in the graph neural network. This step is applied to every edge in the subgraph.

Note that existing graph attention approaches (Cai and Lam, 2020; Shaw *et al.*, 2018; Veličković *et al.*, 2018) generate attention weights for every edge in every layer. However, this way can provide potentially contradicting signals across layers for the same edge, precluding the generation of interpretable pathways. In the contrast, SumGNN adopts a layer-independent attention mechanism, which only depends on the first layer embedding to prune edges. It provides an unequivocal pathway for model explainability. As many biological networks are constructed through text mining where many edges are potentially false positives, this pruning mechanism also allows noise reduction.

(C) The multi-channel integration module

To obtain a powerful representation for drug interaction prediction, we integrate a diverse set of information sources.

Channel 1: Summarized knowledge Using the knowledge summarization approach we described above, we identify a summarized subgraph that is important to input drug pairs. We want to generate the latent representation that leverages this subgraph for the input drug pairs. We integrate it using the following message-passing scheme. For each node v, we compute a relation-aware message weighted by the signal intensity score $\mathbf{b}_v^l$ at layer l using the attention score as

$$\mathbf{b}_v^{(l)} = \sum_{u \in \mathcal{N}_v} \alpha_{u,v}^{(l)} \left(\mathbf{h}_u^{(l-1)} \mathbf{W}_r^{(l)} \right), \tag{2}$$

where $\mathcal{N}_v$ denotes the neighbors of node v in subgraph $\mathcal{G}_{Sub}$, $\mathbf{W}_r^{(l)}$ is the weight matrix to transform hidden representation for node u, v's relation r in layer l. To avoid overfitting, we use basis decomposition (Schlichtkrull *et al.*, 2018) to decompose $\mathbf{W}_r^{(l)}$ into the linear combination of a small number of basis matrices $\{\mathbf{V}_b\}_{b \in B}$ as

$$\mathbf{W}_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} \mathbf{V}_b^{(l)}. \tag{3}$$

Then, we propagate the message $\mathbf{b}_v^{(l)}$ to the updated representation $\mathbf{h}_v^{(l)}$ of node v via

$$\mathbf{h}_v^{(l)} = \mathrm{ReLU}\left(\mathbf{W}_{\mathrm{self}}^{(l)} \mathbf{h}_v^{(l-1)} + \mathbf{b}_v^{(l)} \right), \tag{4}$$

where $\mathbf{W}_{\mathrm{self}}$ is the weight matrix to transform the node embedding itself.

Channel 2: Subgraph features To obtain the embeddings for subgraphs (denoted as $\mathbf{h}_{\mathcal{G}_{Sub}}$), we take the average of all node embeddings in $\mathcal{G}_{Sub}$ at layer l projected by a linear layer as

$$\mathbf{h}_{\mathcal{G}_{Sub}}^{(l)} = \mathrm{Mean}\left(\mathbf{W}_{\mathrm{Sub}} \mathbf{h}_i^{(l)} \right). \tag{5}$$

Channel 3: Drug fingerprint Molecular information such as chemical fingerprints have shown to be powerful predictor of drug interactions (Huang *et al.*, 2020a). Thus, in addition to the network representation, we obtain the Morgan fingerprint $\mathbf{f}_v$ (Rogers and Hahn, 2010), which is a predictive descriptor of drugs, for each drug v. Note that it is infeasible to use this feature as the input node feature in the KG since KG consists of various types of nodes other than drugs (e.g. Side Effect, Disease and genes) and they cannot be represented by Morgan Fingerprints, which lead to inconsistent node features for GNN propagation.

Layer-wise channels aggregation. To assemble various representation generated via each layer, we adopt the layer-aggregation mechanism (Xu *et al.*, 2018). We concatenate node/subgraph embeddings in every layer, i.e. $\mathbf{h}_v = [\mathbf{h}_v^{(1)}, \mathbf{h}_v^{(2)}, \ldots, \mathbf{h}_v^{(L)}]$ and $\mathbf{h}_{\mathcal{G}_{Sub}} = [\mathbf{h}_{\mathcal{G}_{Sub}}^{(1)}, \mathbf{h}_{\mathcal{G}_{Sub}}^{(2)}, \ldots, \mathbf{h}_{\mathcal{G}_{Sub}}^{(L)}]$ where is L is the layer size. To integrate chemical fingerprints, we update the layer-aggregated embedding by concatenation of chemical representation: $\mathbf{h}_v = [\mathbf{h}_v \oplus \mathbf{f}_v]$.

At last, we combine the various channels together to obtain the input drug-pairs representation $\mathbf{h}_{u,v} = [\mathbf{h}_u, \mathbf{h}_v, \mathbf{h}_{\mathcal{G}_{Sub}}]$. To predict the relation, we obtain a prediction probability vector $\mathbf{p}_{u,v}$ where each value in the vector corresponds to a the likelihood of a relation. $\mathbf{p}_{u,v}$ is computed via feeding the drug pair representation to a decoder parameterized by $\mathbf{W}_{\mathrm{pred}}$:

$$\mathbf{p}_{u,v} = \mathbf{W}_{\mathrm{pred}} \mathbf{h}_{u,v}. \tag{6}$$

3.2.1 Training and inference

During training, for multi-class classification task, we adopt the cross entropy loss ℓ_{CE} for each edge (u, r, v) as

$$\ell_{\mathrm{CE}}(u, r, v) = -\sum_{r=1}^{R} \log(\hat{y}_r) \cdot y_r, \tag{7}$$

where $\hat{y}_r = \mathrm{softmax}(\mathbf{p}_{u,v}^r) = \frac{\exp(\mathbf{p}_{u,v}^r)}{\sum_{i=1}^{R} \exp(\mathbf{p}_{u,v}^i)}$ and y_r is the binary indicator if class r is the correct label for u and v. For multi-label classification task, given the edge (u, r, v), we adopt the binary cross entropy loss ℓ_{BCE} as

$$\ell_{\mathrm{BCE}}(u, r, v) = -\log \hat{y}_r - \mathbb{E}_{w \sim P_w(v)} \log(1 - y_r^{u,w}), \tag{8}$$

where (u, r, w) is the sampled negative edge for relation r. This is achieved by replacing node v to node w that is sampled randomly according to a distribution $P_w(v) \propto d_w(v)^{3/4}$ (Mikolov *et al.*, 2013). Then $\hat{y}_r = \mathrm{sigmoid}(\mathbf{p}_{u,v}^r)$, $y_r^{uw} = \mathrm{sigmoid}(\mathbf{p}_{u,w}^r)$ is the prediction score for two edges. Considering all edges, the final loss ℓ in SumGNN is

$$\ell = \sum_{(u,r,v) \in \mathcal{E}} \ell(u, r, v), \tag{9}$$

where ℓ is either Equations (7) or (8) depending on the task type. During training, we learn the model parameter by minimizing the total loss ℓ using stochastic gradient optimizers such as Adam (Kingma and Ba, 2014).

During inference, an unseen node pair u, v's subgraph in the KG is extracted and fed into the same pipeline to calculate the relation vector. For multi-class task, we use the highest probability relation as the predicted relation and for multi-label task, we collect all scores from both positive and negative counterparts for all relations.

4 Experiments

4.1 Experiment setup

Datasets (i) DrugBank dataset (Wishart *et al.*, 2018) contains 1709 drugs (nodes) and 136 351 drug pairs (edges), which are associated with 86 types of pharmacological relations between drugs, such as increase of cardiotoxic activity, decrease of serum concentration, etc. Each drug pair can contain one or two relations. As more than 99.8% of edges have only one edge type (Ryu *et al.*, 2018), we filtered the edge with more than one type in our study. (ii) TWOSIDES (Tatonetti *et al.*, 2012) dataset contains 645 drugs (nodes) and 46 221 drug–drug pairs (edges) with 200 different drug side effect types as labels. For each edge, it may be associated with multiple labels. Following (Dai *et al.*, 2020; Zitnik *et al.*, 2018b), we keep 200 *commonly-occurring* DDI types ranging from Top-600 to Top-800 to ensure every DDI type has at least 900 drug combinations. (iii) For external knowledge base, we use HetioNet (Himmelstein and Baranzini, 2015), which is a large heterogeneous KG merged from 29 public databases. To **ensure no information leakage**, we remove all the overlapping DDI edges between HetioNet and the dataset. In the end, we obtain 33 765 nodes out of 11 types (e.g. gene, disease, pathway, molecular function, etc.) with 1 690 693 edges from 23 relation types.

Baselines We compare our models with several baselines (Further details on baseline methods, implementation and parameters are in the supplementary.).

- MLP (Rogers and Hahn, 2010) uses a two-layer MLP on Morgan fingerprint to directly predict drug interactions.
- Deepwalk (Perozzi *et al.*, 2014) first learns the embeddings for drugs in the network via random walk. Then, it predict the relation for drug pairs via a linear layer over embeddings.
- LINE (Tang *et al.*, 2015) use a one-layer feedforward neural network to learn the embeddings for drugs. Then, it stack a linear layer over embeddings to predict the relation.
- Node2vec (Grover and Leskovec, 2016) first learns the embeddings for drugs in the network. Similar to Deepwalk, it predict the relation via a linear layer over embeddings.

- Decagon (Zitnik *et al.*, 2018b) adopts multi-relational graph convolutional network (Schlichtkrull *et al.*, 2018) on the DDI network for drug interaction prediction.
- GAT (Veličković *et al.*, 2018) uses attention networks to aggregate neighborhood information in DDI network.
- SkipGNN (Huang *et al.*, 2020b) predicts drug interactions by aggregating information from both direct interactions and second-order interactions via two GNNs.
- PRD (Wang, 2017) first use KG embeddings for drugs in the KG, then pass through a linear layer for drug interaction prediction.
- KG-DDI (Karim *et al.*, 2019) first extracts KG embeddings for drugs in the KG, then adopts a Conv-LSTM model using the embeddings for drug interaction prediction.
- GraIL (Teru and Hamilton, 2020) is for inductive relation prediction on KGs, which uses local subgraph.
- KGNN (Lin *et al.*, 2020) samples and aggregates neighborhoods for each node from their local receptives via GNN and with external KG, which achieves the state-of-the-art result on binary DDI prediction problem.

Metrics The task on the DrugBank dataset is a multi-class classification, thus we consider the following metrics:

- F1 Score: average F1 score over *different classes* as F1 Score $= \frac{1}{N}\sum_{k=1}^{N}\frac{2P_k \cdot R_k}{P_k+R_k}$, where N is the # of classes and P_k, R_k is the precision and recall for kth class. Since it gives equal weights for each classes, they are *more sensitive* to the results for classes where samples are fewer.
- Accuracy: Accuracy over all samples Accuracy $= \frac{|Y_k \cap \hat{Y}_k|}{|Y_k|}$, Y_k is the predicted labels at k and $\hat{Y}_k$ are the ground-truth labels.
- Cohen's Kappa (Cohen, 1960) measures the inter-annotator agreement as $\kappa = (p_o - p_e)/(1 - p_e)$, where p_o is the observed agreement (identical to accuracy), p_e is the probability of randomly seeing each class.

The task on the TWOSIDES dataset is a multi-label prediction. We follow (Zitnik *et al.*, 2018b) and consider the following measure. For each side effect type, we calculate the performance individually and use the average performance over all side effects as the final result.

- ROC-AUC is the average area under the receiver operating characteristics curve as $\text{ROC} - \text{AUC} = \sum_{k=1}^{n} \text{TP}_k \Delta \text{FP}_k$, where k is kth true-positive and false-positive operating point $(\text{TP}_k, \text{FP}_k)$.
- PR-AUC is the average area under precision-recall curve $\text{PR} - \text{AUC} = \sum_{k=1}^{n} \text{Prec}_k \Delta \text{Rec}_k$ where k is kth precision/recall operating point $(\text{Prec}_k, \text{Rec}_k)$.
- AP@50 is the average precision at 50, where $\text{AP@k} = \frac{|Y_k \cap \hat{Y}_k|}{|Y_k|}$, Y_k is the predicted labels at k and $\hat{Y}_k$ are the ground-truth labels.

Evaluation strategy. For both datasets, we split it into 7:1:2 as train, development and test set. For the DrugBank dataset, since the label distribution is highly imbalanced, we ensure train/dev/test set contain samples from all classes. For the TWOSIDES dataset, we use the same method in (Zitnik *et al.*, 2018b) to generate negative counterparts for each positive edge by sampling the complement set of positive examples. For every experiment, we conduct five independent runs and select the best performing model based on the loss value on the validation set.

4.2 SumGNN achieves superior predictive performance

We report the performance of our model and all baselines in Table 1. From the result, we find that SumGNN achieves the best performance in DDI prediction on two datasets, accurately predicting the correct DDI pharmacological effect consistently. Particularly, SumGNN has 27.19%, 5.47%, 4.65% absolute increase over the best baseline without KG on three metrics respectively on DrugBank dataset and 2.84%, 2.45%, 4.50% increase on TWOSIDES dataset. Also, SumGNN achieves 5.54%, 1.77%, 1.08% and 1.97%, 2.25%. 2.54% absolute increase over the state-

Table 1. SumGNN achieves the best predictive performance compared to state-of-the-art baselines in DDI prediction

Dataset	Dataset 1: DrugBank			Dataset 2: TWOSIDES		
Classification task	Multi-class			Multi-label		
Methods	F1 Score	Accuracy	Cohen's Kappa	ROC-AUC	PR-AUC	AP@50
MLP (Rogers and Hahn, 2010)	61.10 ± 0.38	82.14 ± 0.33	80.50 ± 0.18	82.60 ± 0.26	81.23 ± 0.14	73.45 ± 0.28
Deepwalk (Perozzi *et al.*, 2014)	24.77 ± 0.40	68.50 ± 0.38	58.44 ± 0.23	88.27 ± 0.09	85.42 ± 0.14	81.09 ± 0.16
LINE (Tang *et al.*, 2015)	30.26 ± 0.45	76.57 ± 0.49	70.91 ± 0.53	91.20 ± 0.34	90.02 ± 0.28	84.07 ± 0.19
Node2Vec (Grover and Leskovec, 2016)	24.92 ± 0.32	71.09 ± 0.40	63.79 ± 0.37	90.66 ± 0.13	88.87 ± 0.23	83.00 ± 0.30
Decagon (Zitnik *et al.*, 2018b)	57.35 ± 0.26	87.19 ± 0.28	86.07 ± 0.08	91.72 ± 0.04	90.60 ± 0.12	82.06 ± 0.45
GAT (Veličković *et al.*, 2018)	33.49 ± 0.36	77.18 ± 0.15	74.20 ± 0.23	91.18 ± 0.14	89.86 ± 0.05	82.80 ± 0.17
SkipGNN (Huang *et al.*, 2020b)	59.66 ± 0.26	85.83 ± 0.18	84.20 ± 0.16	92.04 ± 0.08	90.90 ± 0.10	84.25 ± 0.25
PRD (Wang, 2017)	40.73 ± 0.44	81.52 ± 0.34	78.80 ± 0.36	88.75 ± 0.23	85.26 ± 0.33	79.86 ± 0.26
KG-DDI (Karim *et al.*, 2019)	36.39 ± 0.32	82.48 ± 0.12	78.89 ± 0.27	90.75 ± 0.07	88.16 ± 0.12	83.48 ± 0.05
GraIL (Teru and Hamilton, 2020)	81.31 ± 0.30	89.89 ± 0.24	88.07 ± 0.20	92.89 ± 0.09	91.10 ± 0.19	86.21 ± 0.05
KGNN (Lin *et al.*, 2020)	73.99 ± 0.11	90.89 ± 0.20	89.64 ± 0.24	92.84 ± 0.07	90.78 ± 0.20	86.05 ± 0.12
SumGNN (Ours)	**86.85 ± 0.44**	**92.66 ± 0.14**	**90.72 ± 0.13**	**94.86 ± 0.21**	**93.35 ± 0.14**	**88.75 ± 0.22**
SumGNN-KG	78.35 ± 0.51	89.05 ± 0.36	87.28 ± 0.08	92.62 ± 0.13	90.80 ± 0.40	85.75 ± 0.10
SumGNN-Sum (w/o Summarization)	83.20 ± 0.34	90.83 ± 0.19	90.14 ± 0.10	94.09 ± 0.16	92.55 ± 0.24	87.65 ± 0.24
SumGNN-SF (w/o Subgraph Features)	84.47 ± 0.22	91.88 ± 0.21	90.26 ± 0.19	93.94 ± 0.11	92.45 ± 0.22	87.69 ± 0.08
SumGNN-CF (w/o Chemical Features)	83.57 ± 0.36	91.31 ± 0.17	90.07 ± 0.11	94.35 ± 0.11	92.86 ± 0.20	88.10 ± 0.07
SumGNN-LIA (w/o Layer Independent Attention)	86.54 ± 0.22	92.44 ± 0.30	90.34 ± 0.10	93.92 ± 0.31	92.33 ± 0.19	86.15 ± 0.13

Note: Average and standard deviation of five runs are reported. For these metrics, higher values always indicate better performance.

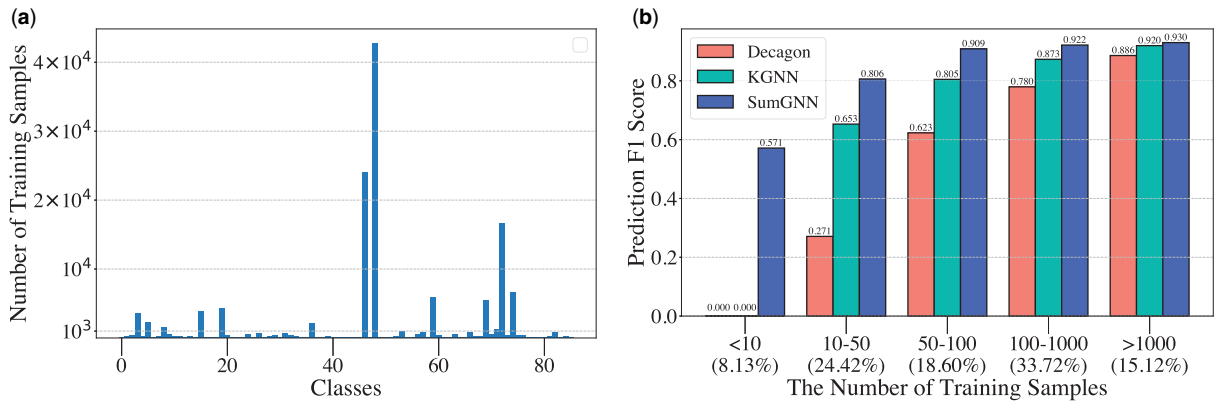

Fig. 2. The dataset statistics and the average F1 score for different relations with various number of training samples on DrugBank dataset. Here, in (**a**), the *x*-axis indicates the index of classes (there are 86 classes in total) and the *y*-axis indicates the number of training samples in the corresponding class. In (**b**), the *x*-axis indicate the range for number of training samples in this group as well as the proportion of classes in this group to all classes and the *y*-axis stands for the average prediction F1 score for the classes in the group

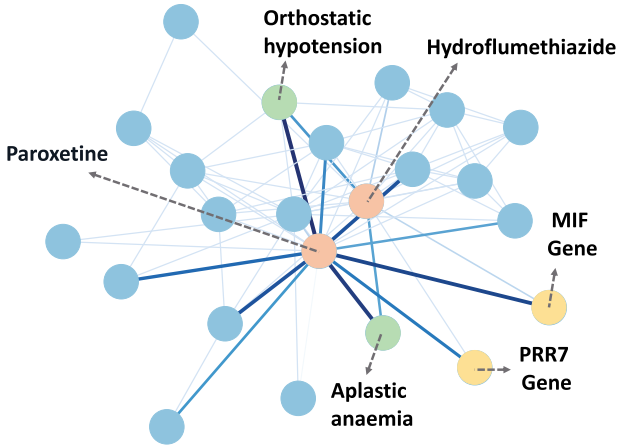

Fig. 3. SumGNN generates a short reasoning path to provide clues for understanding drug interactions. The shade of color indicates the strength of attention weight. Low-weight edges in the extracted subgraph are pruned by SumGNN and SumGNN focus on a sparse set of signal edges and nodes

of-the-art baselines with KG on two datasets. These results clearly verifies the efficacy of our method.

4.3 SumGNN excels at low-data imbalanced relations predictions

We observe that the improvement of SumGNN is much *more significant* on DrugBank than TWOSIDES dataset. We take a closer look at this problem and find that the major difference of these two datasets is that the data distribution of DrugBank is *more imbalanced*—more than 30% of the relation types occurs less than 50 times in the training set while more than 10% of the relation types occurs more than 1000 times, as shown in Figure 2(a).

To examine the model's prediction performance on the size of training data associated with each relation, we first split the relations types into 5 groups with various amounts of training data and then plot the average F1 score of these bins in Figure 2. By comparing the performance of SumGNN against the strongest baseline on Accuracy (KGNN) and the variant of no KG (Decagon), we find that SumGNN can effectively boost the performance when the samples are extremely scarce. When the size of the training samples of the relation is less than 10, both decagon and KGNN cannot give any correct predictions, while our model can still achieve 57.14% F1 score (i.e. choose the correct relation out of the 86 relations). One potential reason is that SumGNN feeds different subgraphs in every GNN propagation, which forms a much-needed inductive bias over unseen subgraphs, such as the ones in the low-data relations. This is in sharp contrast to previous approaches such as KGNN. It also justifies that SumGNN 's knowledge summarization via subgraphs is more effective to harness the external knowledge. In addition, we find SumGNN also brings at least 38.19% improvement on F1 score for relations occuring less than 50 times. This is an important finding since a high overall accuracy does not mean good performance across all relations. The low-data relations are the hardest to predict correctly and we show that SumGNN can be used for these low-data relations whereas other previous models cannot.

4.4 External knowledge improves prediction

We find that the first seven method cannot obtain satisfactory result compared to SumGNN. MLP, Deepwalk, LINE and Node2vec only use shallow embedding layers to learn the features for drugs while do not use graph neural networks for modeling the drug interactions, which have limited expression power. Moreover, although GAT considers the attention on different edges, it fails to consider the multi-relation information, as its performance is also not good. By comparing the latter four methods with the former five methods, we find the use of KG significantly improves DDI performance, highlighting the necessity of combining external knowledge usage with graph neural networks, as it can provide complementary information for DDI task.

4.5 Knowledge summarization module is the best to capture external knowledge

Comparing SumGNN with KGNN and KG-DDI, we show that our model can consistently outperform them on both datasets (more than 4% on average for DrugBank and 2% on average for TWOSIDES), which indicates that simply adopting KG embeddings as well as neighborhood sampling are *insufficient to fully harness the KG information* for DDI prediction. SumGNN provides the best approach to leverage the external KG and also corroborates with our motivation that the use of subgraph reduces noise and irrelevant information. Although GraIL also extracts subgraphs for downstream tasks, it does not use any knowledge summarization techniques, which potentially further eliminating the irrelevant information in the local subgraph. In addition, GraIL merely considers the position information while neglecting the multi-channel features during information propagation.

4.6 Ablation study

To study the usage of KG, we remove the knowledge graph (SumGNN-KG) and perform prediction on the DDI graph. We see SumGNN has 8.5% absolute increase in Macro F1 on DrugBank, highlighting the usefulness of KG. To evaluate the knowledge

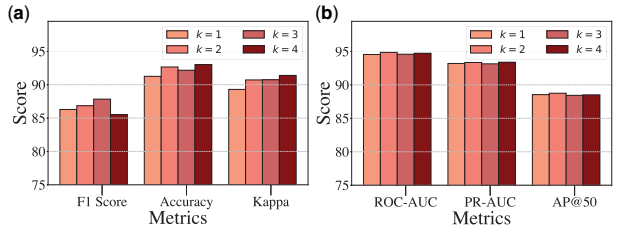

Fig. 4. Model performance given different parameter k

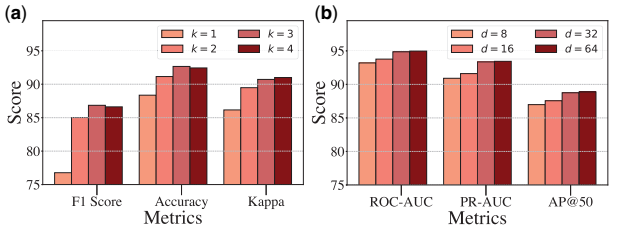

Fig. 5 Model performance given different parameter d

summarization module, we remove the summarization component (SumGNN-Sum) and use the raw local subgraph to predict the outcome. We see SumGNN has 3.65% increase for DrugBank and 2.24% increase for TWOSIDES on Macro F1, suggesting that the summarization further condenses the relevant information and elevate the performance.

To study the effect of multi-channel neural encoding, we compare the result of SumGNN with several variants that remove specific channel information (i.e. subgraph features and chemical features), and we find that these channel information all contribute to the overall performance. Particularly, after removing the subgraph embedding (SumGNN-SF), the Macro F1 drops by 2.38% on DrugBank and 0.92% on TWOSIDES respectively. When removing chemical fingerprint (SumGNN-CF), the performance also degrades 3.28% on Macro F1 for DrugBank, corroborating with the indispensable roles of each of the channel. We also see that by replacing the layer independent attention (SumGNN-LIA) with the previous layer-dependent ones, the performance drops. This suggests that our attention mechanism not only provides interpretability but also increases predictive performance.

4.7 Case study

The usefulness of this model lies in twofolds. First, given the high predictive performance, it can identify novel drug–drug interactions that are flagged high by the model while not in the dataset. Second, using the external knowledge summarization module, we are able to discover signal edges, which provide biological pathways to hint at the potential mechanism of DDIs. We provide a case study of a novel drug pair predicted by the model, Paroxetine and Hydroflumethiazide. Paroxetine is an antidepressant, and Hydroflumethiazide is used to treat hypertension and edema. SumGNN assigns highest probability for the DDI type 'increase of the central nervous system depressant activities'. We then visualize the generated pathway from SumGNN 's summarization scheme in Figure 3. We see that the model significantly reduces irrelevant nodes and edges in the subgraph of the KG and focuses on a sparse set of nodes to make prediction. We examine the nodes that have high signals connection to the target pairs and find literature evidence support. Particularly, the model assigns high weights to two side effects nodes, orthostatic hypotension and aplastic anemia. Orthostatic hypotension refers to a sudden drop in blood pressure when standing up, and aplastic anemia is a condition in which the body stops producing enough new blood cells. Notably, orthostatic hypotension is closely related to multiple system atrophy, a central nervous system problem (Jones *et al.*, 2015). As both drugs incur risk in the side effects, the complication on the central nervous systems could be aggravated when these drugs are taken together, supporting our model prediction. This case study illustrates how to use SumGNN for potential DDI prediction.

4.8 Parameter studies

We study the effect of key parameters. When evaluating one parameter, we fix other parameters to their default values.

• **Effect of the hop of the subgraph k:** Figure 4 shows the result of SumGNN with varying k. From the result, we find that for DrugBank dataset, the performance first increase when k is small. However, when k increases from 3 to 4, we observe slight performance drops on F1 score. These indicate that the larger subgraph can bring more useful information while when k is too large, it may also bring some noise and hurt the performance. For TWOSIDES dataset, we find the result is more stable with different k, indicating even 1-hop subgraph provides adequate information for DDI task.

Table 2. The running time for one epoch of SumGNN for two datasets

Dataset	Size of Hop k				w/o subgraph
	1	2	3	4	
DrugBank	132 s	141 s	178 s	205 s	1279 s
TWOSIDES	62 s	75 s	91 s	102 s	471 s

Note: Note that w/o Subgraph is a variant that directly aggregates the information for all neighbors on KG.

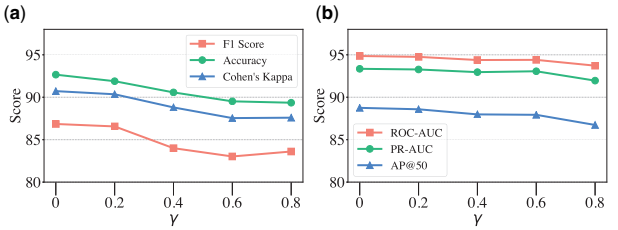

Fig. 6. Model performance given different parameter γ

Moreover, to show how **subgraph formulation drives efficiency**, we compare the training time between SumGNN with varying subgraph size and SumGNN with the entire KG to propagate (See Table 2). We find SumGNN saves 80% of training time via subgraph anchoring, which demonstrates the efficiency of our approach.

• **Effect of the dimension of embeddings d:** Figure 5 exhibits the influence of embedding dimension d. The result indicates that when d is small, increasing d can boost the performance. But when d becomes large, the gain is marginal.

• **Effect of the threshold for summarization γ:** Figure 6 shows the result of SumGNN with different threshold γ, which demonstrates that when γ is small, the performance is rather stable as filtering edges with low score has little effect on the final prediction. This also means SumGNN is able to achieve similar predictive performance while removing many irrelevant edges. However, when γ is large ($\gamma > 0.4$ for DrugBank and $\gamma > 0.6$ for TWOSIDES), it is clear that the performance drops more. In such cases, the summarized graph is more sparse and we might filter out potential useful edges. To sum up, there is a trade-off between the explainability and performance.

5 Conclusion

In this article, we propose a new method SumGNN: *knowledge summarization graph neural network* for multi-typed DDI predictions, which is mainly enabled by a local subgraph module that can

efficiently anchor on relevant subgraphs from a KG, a self-attention based subgraph summarization scheme that can generate reasoning path within the subgraph, and a multi-channel knowledge and data integration module that utilizes massive external biomedical knowledge for significantly improved multi-typed DDI predictions. Experiments on real-world datasets demonstrated the strong performance of SumGNN.

In addition, computational approaches depend heavily on the training data. If the number of training data associated with one specific drug interaction type is low, it is difficult to predict accurately. In contrast to other works, SumGNN is able to generate good performance in low-resource settings. SumGNN is also a general framework and can be adapted to predict any other interactions such as drug-disease interaction. The ability to low-resource learning could also mean to excel at finding drugs for rare diseases.

Funding

This work was in part supported by the National Science Foundation award SCH-2014438, IIS-1418511, CCF-1533768, IIS-1838042, the National Institute of Health award NIH R01 1R01NS107291-01 and R56HL138415.

Conflict of Interest: none declared.

Data Availability

All data and code for SumGNN are available at https://github.com/yueyu1030/SumGNN.

References

Alsentzer,E. *et al.* (2020) Subgraph neural networks. In: *NeurIPS, Virtual.*

Bordes,A. *et al.* (2013) Translating embeddings for modeling multi-relational data. In *NeurIPS*, Lake Tahoe, USA.

Burkhardt,H.A. *et al.* (2019) Predicting adverse drug–drug interactions with neural embedding of semantic predications. *bioRxiv*, 752022.

Cai,D. and Lam,W. (2020) Graph transformer for graph-to-sequence learning. *AAAI*, New York, USA, **34**, 7464–7471.

Celebi,R. *et al.* (2019) Evaluation of knowledge graph embedding approaches for drug–drug interaction prediction in realistic settings. *BMC Bioinformatics*, **20**(1), 1–14.

Chiang,W.-L. *et al.* (2019) Cluster-gcn: an efficient algorithm for training deep and large graph convolutional networks. In: *KDD, Anchorage, USA*.

Cohen,J. (1960) A coefficient of agreement for nominal scales. *Educ. Psychol. Measur.*, **20**, 37–46.

Dai,Y. *et al.* (2020) Wasserstein adversarial autoencoders for knowledge graph embedding based drug–drug interaction prediction. *arXiv*, 07341.

Grover,A. and Leskovec,J. (2016) node2vec: scalable feature learning for networks. In: *KDD*, San Francisco, USA, pp. 855–864.

Gysi,D.M. *et al.* (2020) Network medicine framework for identifying drug repurposing opportunities for covid-19. *arXiv*, 07229.

Himmelstein,D.S. and Baranzini, S.E. (2015) Heterogeneous network edge prediction: a data integration approach to prioritize disease-associated genes. *PLoS Comput. Biol.*, 11, e1004259.

Huang,K. and Zitnik,M. (2020) Graph meta learning via local subgraphs. In: *NeurIPS*, Virtual.

Huang,K. *et al.* (2020a) Caster: predicting drug interactions with chemical substructure representation. *AAAI*, New York, USA, **34**, 702–709.

Huang,K. *et al.* (2020b) SkipGNN: predicting molecular interactions with skip-graph networks. *Scientific Rep.*, **10**(1), 1–16.

Ioannidis,V.N. *et al.* (2020) DRKG – Drug Repurposing Knowledge Graph for Covid-19. https://github.com/gnn4dr/DRKG/.

Jones,P.K. *et al.* (2015) Orthostatic hypotension: managing a difficult problem. *Exp. Rev. Cardiovasc. Ther.*, **13**, 1263–1276.

Karim,M.R. *et al.* (2019) Drug–drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network. In: *ACM-BCB*, Niagara Falls, USA, pp. 113–123.

Kingma,D.P. and Ba,J. (2014) Adam: a method for stochastic optimization. *arXiv, 1412.6980*.

Kipf,T.N. and Welling,M. (2017) Semi-supervised classification with graph convolutional networks. In: *ICLR*, Toulon, France.

Li,P. *et al.* (2015) Large-scale exploration and analysis of drug combinations. *Bioinformatics*, **31**, 2007–2016.

Liang,C. *et al.* (2020) Bond: bert-assisted open-domain named entity recognition with distant supervision. In: *KDD*, Virtual, pp. 1054–1064.

Lin,X. *et al.* (2020) KGNN: knowledge graph neural network for drug–drug interaction prediction. In: *IJCAI*, Virtual.

Mikolov,T. *et al.* (2013) Distributed representations of words and phrases and their compositionality. In: *NeurIPS*, Lake Tahoe, USA.

Percha,B. and Altman,R.B. (2013) Informatics confronts drug–drug interactions. *Trends Pharmacol. Sci.*, **34**, 178–184.

Perozzi,B. *et al.* (2014) Deepwalk: online learning of social representations. In: *KDD*, New York, USA.

Rogers,D. and Hahn,M. (2010) Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, **50**, 742–754.

Rotmensch,M. *et al.* (2017) Learning a health knowledge graph from electronic medical records. *Sci. Rep.*, **7**, 5994.

Ryu,J.Y. *et al.* (2018) Deep learning improves prediction of drug–drug and drug–food interactions. *Proc. Natl. Acad. Sci. USA*, **115**, E4304–4311.

Schlichtkrull,M. *et al.* (2018) Modeling relational data with graph convolutional networks. In: *ESWC*, Crete, Greece.

Shang,J. *et al.* (2019) Pre-training of graph augmented transformers for medication recommendation. In: *IJCAI*, Macao, China.

Shaw,P. *et al.* (2018) Self-attention with relative position representations. In: *NAACL*, New Orleans, USA.

Shi,H. *et al.* (2020). Predicting origin-destination flow via multi-perspective graph convolutional network. In: *ICDE*, Virtual.

Srinivasa,R.S. *et al.* (2020) Fast graph attention networks using effective resistance based graph sparsification. *arXiv*, 08796.

Su,C. *et al.* (2020) Network embedding in biomedical data science. *Brief. Bioinf.*, **21**, 182–197.

Sun,Z. *et al.* (2018) Recurrent knowledge graph embedding for effective recommendation. In: *RecSys*, Vancouver, Canada, pp. 297–305.

Takeda,T. *et al.* (2017) Predicting drug–drug interactions through drug structural similarities and interaction networks incorporating pharmacokinetics and pharmacodynamics knowledge. *Journal of Cheminformatics*, **9**, 2688–2694.

Tang,J. *et al.* (2015) Line: large-scale information network embedding. In: *WWW*, Florence, Italy, pp. 1067–1077.

Tatonetti,N.P. *et al.* (2012) Data-driven prediction of drug effects and interactions. *Sci. Transl. Med.*, **4**, 125ra31.

Teru,K. and Hamilton,W. (2020) Inductive relation prediction on knowledge graphs. In: *ICML*, Virtual.

Trouillon,T. *et al.* (2016) Complex embeddings for simple link prediction. In: *ICML*, New York, USA.

Tzeng,R.-C. and Wu,S.-H. (2019) Distributed, egocentric representations of graphs for detecting critical structures. In: *ICML*, Long Beach, USA.

Veličković,P. *et al.* (2018) Graph attention networks. In: *ICLR*, Vancouver, Canada.

Veličković,P. *et al.* (2019) Deep graph infomax. In: *ICLR*, New Orleans, USA.

Wang,G. *et al.* (2018) Label-free distant supervision for relation extraction via knowledge graph embedding. In: *EMNLP*, Brussels, Belgium.

Wang,H. *et al.* (2019a) Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In: *KDD*, Anchorage, USA, pp. 968–977.

Wang,X. *et al.* (2019b) KGAT: knowledge graph attention network for recommendation. In: *KDD*, Anchorage, USA.

Wang,M. (2017) Predicting rich drug–drug interactions via biomedical knowledge graphs and text jointly embedding. *arXiv Preprint arXiv*, **1712**, 08875.

Wishart,D.S. *et al.* (2018) Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic Acids Res.*, **46**, D1074–D1082.

Xu,K. *et al.* (2018) Representation learning on graphs with jumping knowledge networks. In: *ICML*, Stockholm, Sweden.

Yu,D. *et al.* (2020a) Generalized multi-relational graph convolution network. *arXiv*, 07331.

Yu,Y. *et al.* (2020b) Semantic-aware spatio-temporal app usage representation via graph convolutional network. In: *IMWUT*, Virtual.

Zeng,H. *et al.* (2020) Graphsaint: graph sampling based inductive learning method. In: *ICLR*, Virtual.

Zhang,M. and Chen,Y. (2018) Link prediction based on graph neural networks. In: *NeurIPS*, Montreal, Canada.

Zitnik,M. *et al.* (2018a) *BioSNAP Datasets: Stanford Biomedical Network Dataset Collection*. http://snap.stanford.edu/biodata.

Zitnik,M. *et al.* (2018b) Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, **34**, i457–i466.