

# Emerging Drug Interaction Prediction Enabled by Flow-based Graph Neural Network with Biomedical Network

Yongqi Zhang<sup>1</sup>, Quanming Yao<sup>2\*</sup>, Ling Yue<sup>2</sup>, Xian Wu<sup>3</sup>, Ziheng Zhang<sup>3</sup>, Zhenxi Lin<sup>3</sup> and Yefeng Zheng<sup>3</sup>

<sup>1</sup>4Paradigm Inc., Beijing, China.

<sup>2</sup>Department of Electronic Engineering, Tsinghua University, Beijing, China.

<sup>3</sup>Tencent Jarvis Lab, Shenzhen, China.

\*Corresponding author(s). E-mail(s): [qyaoaa@tsinghua.edu.cn](mailto:qyaoaa@tsinghua.edu.cn);

Contributing authors: [zhangyongqi@4paradigm.com](mailto:zhangyongqi@4paradigm.com); [lingyue@tsinghua.edu.cn](mailto:lingyue@tsinghua.edu.cn);

[kevinxwu@tencent.com](mailto:kevinxwu@tencent.com); [zihengzhang@tencent.com](mailto:zihengzhang@tencent.com); [chalerislin@tencent.com](mailto:chalerislin@tencent.com);

[yefengzheng@tencent.com](mailto:yefengzheng@tencent.com);

## Abstract

Emerging drugs offer new possibilities for treating and alleviating diseases, but their potential interactions with existing drugs need to be carefully assessed for clinical and developmental purposes. Accurately predicting drug-drug interactions (DDI) for emerging drugs with computational methods can clinically improve patient care and contribute to efficient drug development. However, most existing deep learning methods require large amounts of DDI data that are scarce for emerging drugs. In this paper, we propose EmerGNN, a graph neural network (GNN) that can effectively predicts interactions for emerging drugs by leveraging the rich information in biomedical networks. EmerGNN learns pairwise representations of drugs by extracting the paths between drug pairs, propagating information from one drug to the other, and incorporating the relevant biomedical concepts on the paths. Our method weights the different edges on the biomedical network to extract relevant information for DDI prediction and provides interpretable evidence for the prediction using attention weights on the edges. Extensive experiments demonstrate that EmerGNN outperforms existing methods in predicting interactions for emerging drugs, and the learned concepts on the biomedical network are interpretable.

**Keywords:** Drug-drug interaction, emerging drug, biomedical network, graph neural network

## 1 Introduction

Owing to science development and regulatory reforms, a large quantity of emerging drugs are being developed all over the world, especially to treat rare, serious or life-threatening diseases ([Su et al, 2022](#); [Ledford, 2022](#)). These drugs are new or novel substances that have not been widely used or regulated before, thus can have unknown

or unpredictable risks. For example, hundreds of drugs have been developed for COVID-19, but the FDA (Food and Drug Administration) has only recommended six drugs for the treatment of COVID-19 as of May 2023<sup>1</sup>, including dexamethasone and hydrocortisone. Besides, the clinical

---

<sup>1</sup><https://www.fda.gov/drugs/emergency-preparedness-drugs-coronavirus-covid-19-drugs>

deployment of new emerging drugs is prudent and slow. Hence, identifying drug-drug interactions (DDIs) for emerging drugs is of great importance. Due to the time-consuming development cycles of in vivo experiments, computational techniques especially machine learning approaches have been developed to speedup the discovery of potential DDIs (Percha and Altman, 2013; Vilar et al, 2014; Tanvir et al, 2021; Yu et al, 2021). However, with little information collected from limited clinical trials on these emerging drugs, the problem of unexpected polypharmacy or side effects becomes severe and hard to detect (Letinier et al, 2019; Jiang et al, 2022).

Early works for DDI prediction relied on fingerprints (Rogers and Hahn, 2010) or hand-designed features (Vilar et al, 2014; Dewulf et al, 2021) to indicate interactions based on both drugs' properties. Even though these methods can directly work on emerging drugs in a cold-start setting (Liu et al, 2022; Dewulf et al, 2021), they lack expressiveness as they are shallow and ignore the mutual information between drugs. DDI facts can naturally be represented as a graph where each node corresponds to a drug and each edge represents an interaction between a pair of drugs. Graph learning methods can be applied to learn drug embeddings for prediction (Yao et al, 2022), but they rely on historical interactions, thus cannot well address the problem of emerging drugs, whose interaction data is scarce.

Apart from solely learning from the DDI interactions, large biomedical networks can be incorporated as side information for DDI prediction (Zitnik et al, 2018; Karim et al, 2019; Tanvir et al, 2021; Huang et al, 2020; Yu et al, 2021; Lin et al, 2020; Ren et al, 2022). These networks, e.g., Het-ioNet (Himmelstein et al, 2017), organize facts into a directed multi-relational graph with millions of edges, recording the relationships between biomedical concepts, such as genes, diseases, and drugs, in biomedicine and healthcare. Tanvir et al (2021) used hand-designed meta-path from the biomedical network for prediction. Karim et al (2019) learned embeddings from the biomedical network and used a deep network to do DDI prediction. There are several methods (Zitnik et al, 2018; Huang et al, 2020; Yu et al, 2021; Lin et al, 2020; Ren et al, 2022) applying graph neural network (GNN) (Kipf and Welling, 2016; Gilmer

et al, 2017) to obtain expressive node embeddings by aggregating the topological structure from the biomedical network and the embeddings of drugs. However, existing methods aggregate information in the same style over the entire biomedical network without specially considering the emerging drugs, leading to poor performance in prediction DDI for emerging drugs.

In this paper, we propose to learn from the biomedical concepts connecting target drugs pairs in the large biomedical networks to predict DDI for emerging drugs. Our intuition is that while emerging drugs may not have sufficient interactions in the DDI network, they often share the same biochemical concepts used in the drug development with existing drugs, like the same targeted genes or diseases. Thus, we exploit the related subgraphs from the large biomedical networks for the given drug pairs. However, properly leveraging biomedical networks can be challenging as these networks are not specially developed for emerging drugs and the mismatch of objectives can lead the machine learning models to learn distractive knowledge.

To predict DDI for emerging drugs in an accurate and interpretable way, we propose EmerGNN, a GNN method that learns pair-wise representations between drugs by integrating the biomedical entities and relations connecting them. A flow-based GNN architecture is designed to extract the paths connecting drug pairs, trace from an emerging drug to an existing drug, and integrate information of the biomedical concepts on the paths. In this approach, the shared information in both biomedical network and interaction network can be utilized. To extract relevant information in the biomedical network for DDI prediction, we weight the different types of relations on the biomedical network. The edges with larger weights on the paths are helpful for interpretation. Compared with other GNN-based methods, EmerGNN propagates on the local subgraph around the drug pair to be predicted and can better discover directional information flow within the biomedical network. In summary, our main contributions are as follows:

- Building upon a biomedical network, we develop an effective deep learning method that predicts interactions for emerging drugs accurately.
- We propose EmerGNN, a GNN-based method that learns pair-wise representations of drug

pairs to predict DDIs for emerging drugs by integrating the relevant biomedical concepts connecting them.

- Extensive experiments show that EmerGNN is more effective in predicting interactions for emerging drugs. The learned concepts on the biomedical network are interpretable.
- EmerGNN’s strong prediction ability has the potential to clinically improve patient care and contribute to more efficient drug development processes.

## 2 Results

### 2.1 Problem Formulation

There are two tasks in DDI prediction for emerging drugs (Dewulf et al., 2021; Liu et al., 2022; Yu et al., 2022). In the S1 setting, the task is to determine which interaction type an emerging drug and an existing drug will have. In the S2 setting, the task is to determine the interaction type between two emerging drugs. Let  $\mathcal{V}_D$  be the set of existing drugs,  $\mathcal{R}_I$  be the set of pharmacological drug-drug interaction (DDI) types, and  $\mathcal{N}_D = \{(u, i, v) : u, v \in \mathcal{V}_D, i \in \mathcal{R}_I\}$  be the drug interaction network. Classically, the DDI prediction task is to learn a predictor  $p : \mathcal{V}_D \times \mathcal{V}_D \rightarrow \mathcal{R}_I$  that can accurately predict the interaction type  $i \in \mathcal{R}_I$  between drug pairs  $(u, v) \in \mathcal{V}_D \times \mathcal{V}_D$ . Here, we consider the DDI prediction for emerging drug, where either  $u$  or  $v$  is an emerging drug. Thus, there are no interactions between such  $u$  and  $v$  in  $\mathcal{N}_D$ .

To provide connection between emerging drugs and existing drugs, a large biomedical network is provided, e.g. HetioNet (Himmelstein et al., 2017), with many entities and relations related to biomedical concepts. We denote the set of biomedical entities as  $\mathcal{V}_B$ , the relation types among entities as  $\mathcal{R}_B$ , and the edges as  $\mathcal{N}_B = \{(h, r, t) : h, t \in \mathcal{V}_B, r \in \mathcal{R}_B\}$ . We assume that all the emerging drugs are connected to some entities in  $\mathcal{V}_B$  such that we can infer their properties from existing drugs by leveraging information in the biomedical network.

### 2.2 Framework of EmerGNN

The proposed EmerGNN framework is shown in Figure 1. Given the DDI network  $\mathcal{N}_D$  and

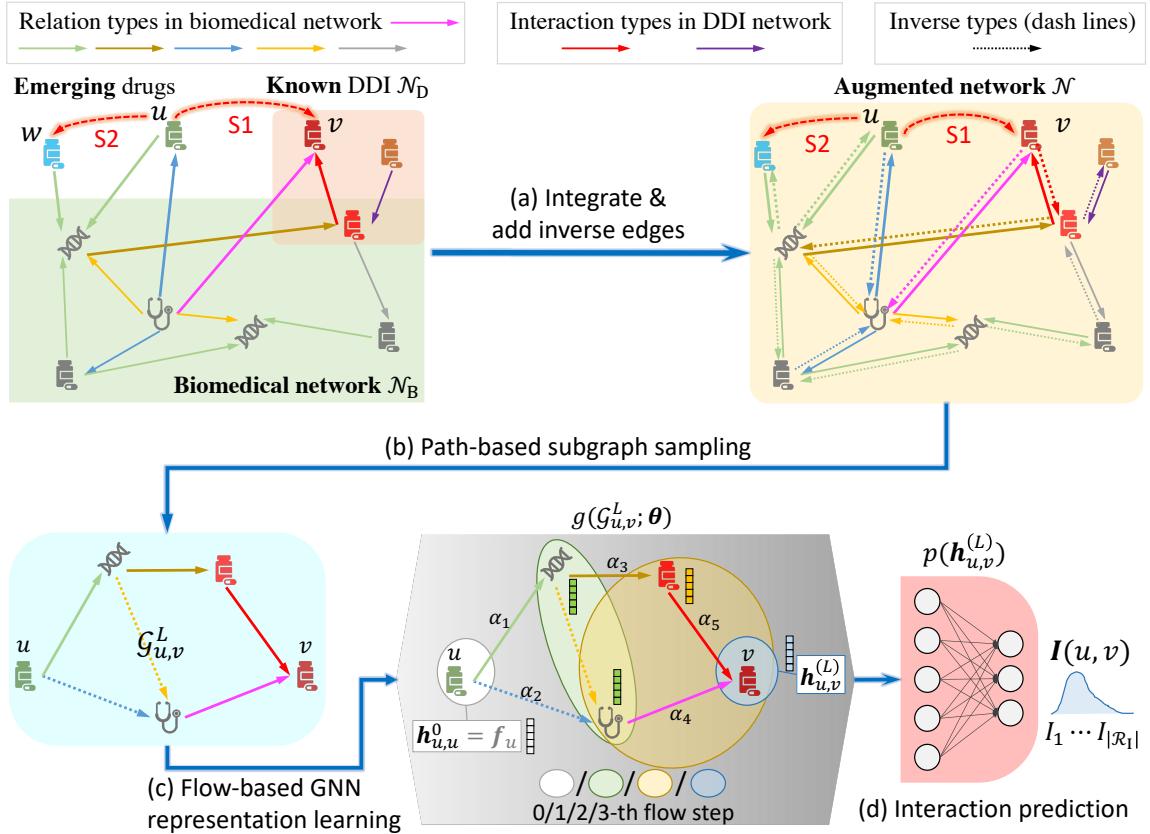
biomedical network  $\mathcal{N}_B$ , we firstly integrate the two networks to enable the existing and emerging drugs connected by concepts, like proteins, diseases or other drugs, in  $\mathcal{N}_B$ . Then, following (Vashisht et al., 2019; Yu et al., 2021), we add inverse edges by introducing inverse types for each relation and interaction types. The two steps generate an augmented network  $\mathcal{N}$  where the drugs and biomedical entities can have better communication. For a target pair of drugs to be predicted (e.g. an emerging drug  $u$  and an existing drug  $v$ ), we extract all the paths with length no longer than  $L^2$  between  $u$  and  $v$ , then combine the paths to form a path-based subgraph  $\mathcal{G}_{u,v}^L$ . All the edges in  $\mathcal{G}_{u,v}^L$  are pointed away from  $u$  and towards  $v$ . A flow-based GNN  $g(\cdot)$  is applied on  $\mathcal{G}_{u,v}^L$  to trace the drug features  $\mathbf{h}_{u,u}^0 = \mathbf{f}_u$  (e.g. fingerprints) along the biomedical edges and integrate essential information along the path. Note that in each iteration  $\ell$ , the GNN flows to the drug-specific entities that are  $\ell$ -steps away from drug  $u$  and  $(L - \ell)$ -steps away from drug  $v$  in the augmented network. An attention mechanism is applied on the edges in  $\mathcal{G}_{u,v}^L$  to adjust their importance. The flow-based GNN iterates for  $L$  steps and returns the pair-wise representation  $\mathbf{h}_{u,v}^{(L)}$ . Finally,  $\mathbf{h}_{u,v}^{(L)}$  is fed to a linear classifier  $p(\cdot)$  to predict the interaction type between  $u$  and  $v$ .

### 2.3 Evaluation Strategies and Metrics

We study the DDI prediction problem based on two public datasets DrugBank (Wishart et al., 2018) and TWOSIDES (Tatonetti et al., 2012) for the pharmacological effects study. The drug set  $\mathcal{V}_D$  is split into three disjoint parts  $\mathcal{V}_{D\text{-train}}$ ,  $\mathcal{V}_{D\text{-valid}}$ ,  $\mathcal{V}_{D\text{-test}}$  for training, validation, and testing, respectively, with a ratio of 7:1:2. The drugs in  $\mathcal{V}_{D\text{-valid}}$  and  $\mathcal{V}_{D\text{-test}}$  are regarded as emerging drugs for validation and testing, respectively. The training set of interactions  $\mathcal{N}_{D\text{-train}}$  contains all the drugs in  $\mathcal{V}_{D\text{-train}}$ . In the S1 setting,  $\mathcal{N}_{D\text{-valid}}$  consists of interactions between drug pairs, each with one drug in  $\mathcal{V}_{D\text{-train}}$  and the other in  $\mathcal{V}_{D\text{-valid}}$ . For each drug pair in  $\mathcal{N}_{D\text{-test}}$ , one drug is in  $\mathcal{V}_{D\text{-train}} \cup \mathcal{V}_{D\text{-valid}}$  and the other is in  $\mathcal{V}_{D\text{-test}}$ . In the S2 setting,  $\mathcal{N}_{D\text{-valid}}$  contains interactions

---

<sup>2</sup>The value of  $L$  is a hyper-parameter to be selected as discussed in Supplementary Section A.3.



**Fig. 1** Overview of the EmerGNN framework. We have a DDI network of existing drugs and a large biomedical network providing side information for the drugs. To predict the interaction type in the S1 setting between an existing drug (e.g.  $u$  in green) and an existing drug (e.g.  $v$  in red) or the S2 setting between two emerging drugs (e.g.  $u$  in green and  $w$  in blue), (a) we integrate the DDI network and biomedical network and add edges with inverse types to obtain an augmented network; (b) we then extract all the paths from  $u$  to  $v$  to construct a path-based subgraph  $\mathcal{G}_{u,v}^L$ ; (c) a flow-based GNN with attention mechanism is applied on  $\mathcal{G}_{u,v}^L$  to trace the drug features  $\mathbf{f}_u$  over essential information in the augmented network for  $L$  steps, resulting in a pair-wise representation  $\mathbf{h}_{u,v}^{(L)}$ ; (d) a simple classifier  $p(\cdot)$  is used to predict the interaction type  $i \in \mathcal{R}_I$  between  $u$  and  $v$ .

with both drugs in  $\mathcal{V}_{\text{D-valid}}$ , and  $\mathcal{N}_{\text{D-test}}$  contains interactions with both drugs in  $\mathcal{V}_{\text{D-test}}$ . For the biomedical network, the edges in  $\mathcal{N}_{\text{B-train}}$  for training do not contain any drugs in  $\mathcal{V}_{\text{D-valid}}$  and  $\mathcal{V}_{\text{D-test}}$ ;  $\mathcal{N}_{\text{B-valid}}$  for validation does not contain any drugs in  $\mathcal{V}_{\text{D-test}}$ ;  $\mathcal{N}_{\text{B-test}} = \mathcal{N}_{\text{B}}$  for testing includes the edges in the biomedical network. For each setting, we generate five random splits of the data with different random seeds. The details of dataset generation are provided in Supplementary Section A.1.

For the DrugBank dataset, there is at most one interaction type between any pair of drugs, the task is to predict the exact type  $i \in \mathcal{R}_I$  in a multi-type classification setting. Three metrics are utilized: “F1-score” (macro) that averages all

the per-class F1-scores; “Accuracy” that measures the percentage of correctly predicted classes; and “Cohen’s Kappa” (Cohen, 1960) that assesses the agreement between model accuracy and random guessing accuracy. F1-score is used as the primary metric as it can better handle the problem of class imbalance.

For the TWO SIDES dataset, there may be multiple interaction types between a pair of drugs, and the task is to predict whether a pair of drugs will have a certain interaction type under a binary classification setting. Three metrics are used: “PR-AUC” that is the area under the precision-recall curve; “ROC-AUC” that measures the area under the curve of true-positive vs. false-positive; and

“Accuracy” that measures the percentage of correctly predicted triplets. PR-AUC is used as the primary metric as we care more about the positive classes.

The more details of evaluation metrics are provided in Supplementary Section A.2. The hyper-parameters used for model evaluation are selected according to the primary metric performance on the validation dataset, with more details provided in Supplementary Section A.3.

## 2.4 Comparison of DDI Prediction Performance

We compare **EmerGNN** with four types of methods, which are popularly used in DDI prediction. (i) Methods with drug features (*DF*). [Rogers and Hahn \(2010\)](#) learns a multi-layer perception (denoted as *MLP*) that mapped the fingerprints of drugs to their interaction type; [Vilar et al \(2014\)](#) uses the similarity between fingerprint features (denoted as *Similarity*) for prediction based on the assumption that similar drugs have similar interaction types; [CSMDDI \(Liu et al, 2022\)](#) learns a function to map drugs’ attributes to representations for DDI prediction in a cold-start setting; and [STNN-DDI \(Yu et al, 2022\)](#) uses tensor factorization technique for cold-start DDI prediction with drugs’ SMILES attributes. (ii) A method based on graph features (*GF*), i.e., [HIN-DDI \(Tanzir et al, 2021\)](#), which extracts meta-paths on the biomedical network and predicts the interaction type based on the meta-paths. (iii) Methods that predict interactions by learning drug embeddings (*Emb*), i.e., [MSTE \(Yao et al, 2022\)](#) and [KG-DDI \(Karim et al, 2019\)](#). (iv) Graph neural network (*GNN*)-based methods. [CompGCN \(Vashishth et al, 2019\)](#) uses GNN to learn high-order embeddings of entities from their neighbors in a knowledge graph; [Decagon \(Zitnik et al, 2018\)](#) has similar model structure as [CompGCN](#), but only uses three types of entities, i.e., drug, protein and disease; [KGNN \(Lin et al, 2020\)](#) randomly samples a subset of edges when aggregating the neighbors; [SumGNN \(Yu et al, 2021\)](#) uses GNN to summarize knowledge in the subgraphs covering the drug pairs; [DeepLGF \(Ren et al, 2022\)](#) fuses multiple types of information including SMILES, DDI interactions, and representations aggregated by a GNN.

For each method, we run for five times on the five-fold datasets and report the mean value with

standard deviation on the testing data in Table 1. All the methods run on a single Nvidia RTX 3090 GPU with 24GB memory, with implementation details provided in Supplementary Section A.4.

### Results for *S1* setting

The empirical comparison is provided in Table 1(a). The *Emb* type methods, especially [MSTE](#), are the worst for emerging drug prediction. Considering that the emerging drugs are not available during training, their embeddings are not updated, thus [MSTE](#) will have no knowledge about the emerging drugs. [KG-DDI](#) is better than [MSTE](#) as the drug embeddings can be updated with information in the biomedical network. With specially designed training schemes in a cold-start setting, [CSMDDI](#) and [STNN-DDI](#) are better than [MLP](#) on the DrugBank dataset, but they do not perform well on [TWOSIDES](#) which has more interaction types. [HIN-DDI](#) is better than [MLP](#), showing that the graph features from biomedical network can also benefit the DDI prediction. The deep *GNN*-based methods may not perform better than *DF* methods on DrugBank. As we will show in Section 2.6, similarity property is important for emerging drug prediction, but the *GNN*-based methods cannot well capture this pattern. [CompGCN](#), [Decagon](#) and [KGNN](#) are comparable with each other since they have similar message functions in the GNN architecture design. [SumGNN](#) constrains the message passing in the enclosing subgraph between drug pairs, thus the information is more focused. [DeepLGF](#) is the best GNN-based method by fusing different information from multiple sources, taking the advantage of both drug features and graph features.

Overall, **EmerGNN** significantly outperforms all the compared methods as indicated by the small p-values. First, by learning paths between emerging and existing drugs, it can capture the graph features, whose importance has been verified by the *GF* method [HIN-DDI](#). Second, different from [CompGCN](#), [Decagon](#), [KGNN](#), and [DeepLGF](#), the importance of edges can be weighted such that it can implicitly learn the similarity properties (as will be demonstrated in Section 2.6). Third, with the specially designed path-based subgraph and flow-based GNN architecture, our method can capture more essential information from the biomedical network (see a detailed comparison in Section 2.7),

**Table 1** Comparison of different methods on the DDI prediction task. “DF” is short for “Drug Feature”; “GF” is short for “Graph Feature”; “Emb” is short for “Embedding”; and “GNN” is short for “Graph Neural Network”. The evaluation metrics are presented in percentage (%) with best values in boldface and the second best underlined. p-values are computed under t-testing of EmerGNN over the second best. Methods leveraging a biomedical network are indicated by star \*.

(S1): DDI prediction between emerging drug and existing drug.

Datasets		DrugBank			TWOSIDES		
Type	Methods	F1-Score	Accuracy	Kappa	PR-AUC	ROC-AUC	Accuracy
DF	MLP (Rogers and Hahn, 2010)	21.1±0.8	46.6±2.1	33.4±2.5	81.5±1.5	81.2±1.9	76.0±2.1
	Similarity (Vilar et al, 2014)	43.0±5.0	51.3±3.5	44.8±3.8	56.2±0.5	55.7±0.6	53.9±0.4
	CSMDDI (Liu et al, 2022)	45.5±1.8	<u>62.6</u> ±2.8	55.0±3.2	73.2±2.6	74.2±2.9	69.9±2.2
	STNN-DDI (Yu et al, 2022)	39.7±1.8	56.7±2.6	46.5±3.4	68.9±2.0	68.3±2.6	65.3±1.8
GF	HIN-DDI* (Tanvir et al, 2021)	37.3±2.9	58.9±1.4	47.6±1.8	<u>81.9</u> ±0.6	<u>83.8</u> ±0.9	<u>79.3</u> ±1.1
Emb	MSTE (Yao et al, 2022)	7.0±0.7	51.4±1.8	37.4±2.2	64.1±1.1	62.3±1.1	58.7±0.7
	KG-DDI* (Karim et al, 2019)	26.1±0.9	46.7±1.9	35.2±2.5	79.1±0.9	77.7±1.0	60.2±2.2
GNN	CompGCN* (Vashisht et al, 2019)	26.8±2.2	48.7±3.0	37.6±2.8	80.3±3.2	79.4±4.0	71.4±3.1
	Decagon* (Zitnik et al, 2018)	24.3±4.5	47.4±4.9	35.8±5.9	79.0±2.0	78.5±2.3	69.7±2.4
	KGNN* (Lin et al, 2020)	23.1±3.4	51.4±1.9	40.3±2.7	78.5±0.5	79.8±0.6	72.3±0.7
	SumGNN* (Yu et al, 2021)	35.0±4.3	48.8±8.2	41.1±4.7	80.3±1.1	81.4±1.0	73.0±1.4
	DeepLGF* (Ren et al, 2022)	39.7±2.3	60.7±2.4	51.0±2.6	81.4±2.1	82.2±2.6	72.8±2.8
	EmerGNN*	<b>62.0</b> ±2.0	<b>68.6</b> ±3.7	<b>62.4</b> ±4.3	<b>90.6</b> ±0.7	<b>91.5</b> ±1.0	<b>84.6</b> ±0.7
<b>p-value</b>		8.9E-7	0.02	0.02	1.6E-6	6.0E-8	3.5E-5

(S2): DDI prediction between two emerging drugs.

Datasets		DrugBank			TWOSIDES		
Type	Methods	F1-Score	Accuracy	Kappa	PR-AUC	ROC-AUC	Accuracy
DF	CSMDDI (Liu et al, 2022)	19.8±3.1	37.3±4.8	22.0±4.9	55.8±4.9	57.0±6.1	55.1±5.2
	GF HIN-DDI* (Tanvir et al, 2021)	8.8±1.0	27.6±2.4	13.8±2.4	<u>64.8</u> ±2.3	<u>58.5</u> ±1.6	<u>59.8</u> ±1.4
	Emb KG-DDI* (Karim et al, 2019)	1.1±0.1	32.2±3.6	0.0±0.0	53.9±3.9	47.0±5.5	50.0±0.0
	GNN DeepLGF* (Ren et al, 2022)	4.8±1.9	31.9±3.7	8.2±2.3	59.4±8.7	54.7±5.9	54.0±6.2
EmerGNN*		<b>25.0</b> ±2.8	<b>46.3</b> ±3.6	<b>31.9</b> ±3.8	<b>81.4</b> ±7.4	<b>79.6</b> ±7.9	<b>73.0</b> ±8.2
<b>p-value</b>		0.02	0.01	0.01	1.4E-3	3.9E-4	7.8E-3

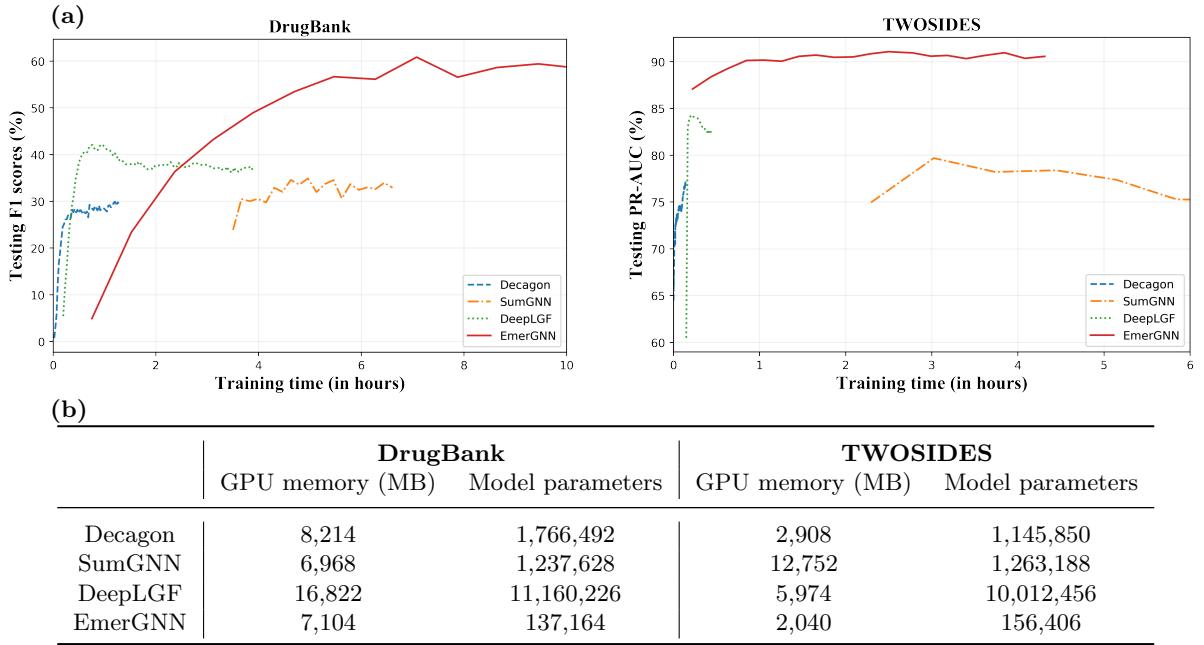
thus outperforms CompGCN and SumGNN as well. We provide more analysis between the subgraphs used by SumGNN and EmerGNN in Supplementary Section B.4.

### Results for S2 setting

In the bottom part of Table 1, we present the results of the top-performing models for predicting interactions between two emerging drugs in the S2 setting. This setting is more challenging than the S1 setting since both drugs are new with sparser information. Our comparison of the two tables reveals that the S2 setting is indeed more difficult, with lower prediction accuracy than the S1 setting. While KG-DDI and DeepLGF performed well in the S1 setting, they struggled with the S2 setting due to the need to learn representations of both drugs effectively. Conversely, CSMDDI and HIN-DDI

performed more consistently with the S1 setting, CSMDDI ranking second in DrugBank and HIN-DDI ranking second in TWOSIDES. This may be due to their simple models but effective features. In comparison, EmerGNN significantly outperforms all the baselines by aggregating essential information from the biomedical network.

We also provide the results of an additional S0 setting, which predicts interactions between two existing drugs, in Supplementary Section C.1. In the following sections, we investigate thoroughly why EmerGNN has superior performance for DDI prediction.



**Fig. 2** Complexity analysis of different GNN-based methods in the S1 setting. (a) training curves; (b) GPU memory footprint and the model parameters.

## 2.5 Analyzing the computational complexity

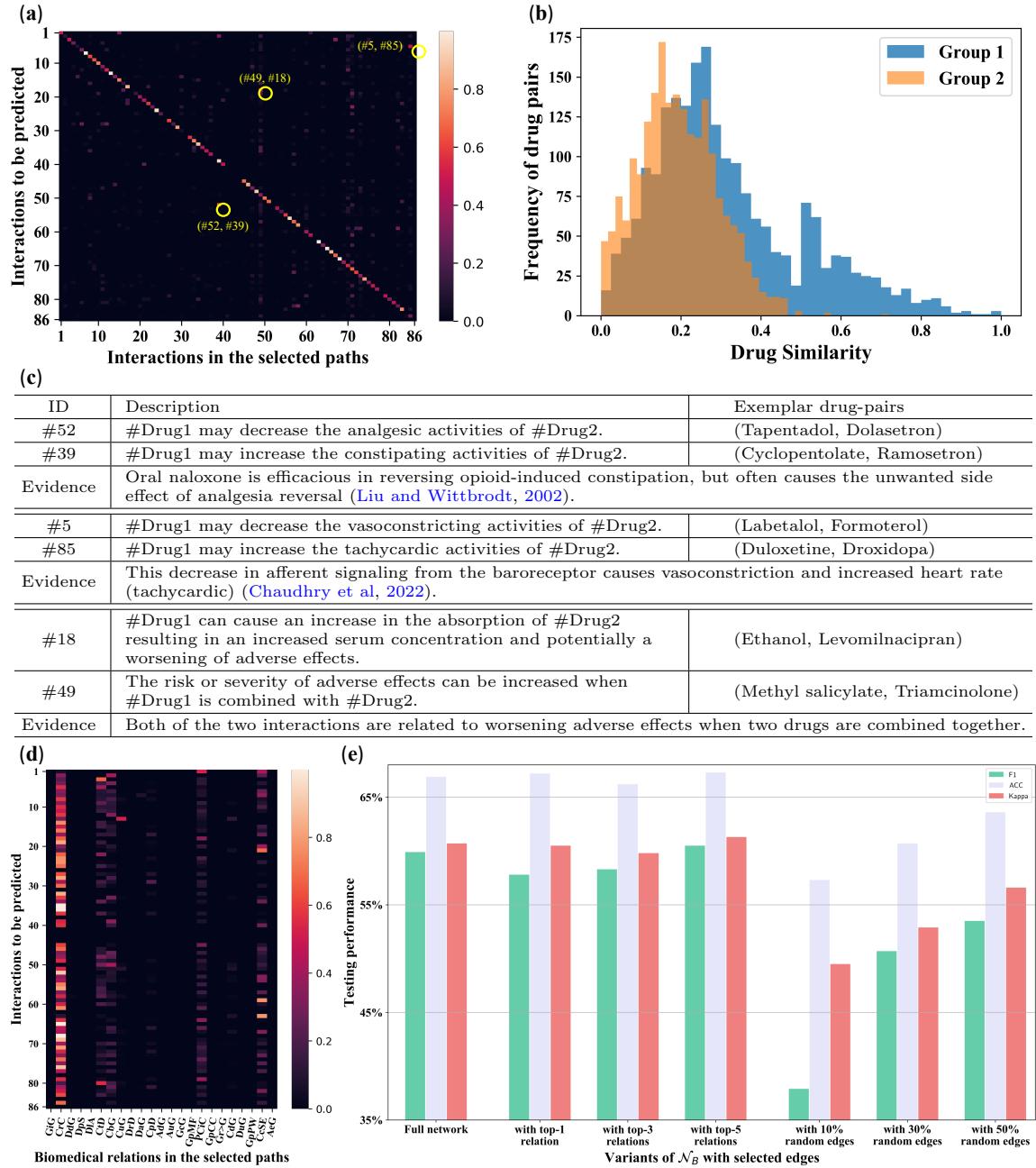
We analyze the computational complexity of EmerGNN in this section. Since EmerGNN learns pair-wise representations for each drug pair, the computation complexity is higher than the other GNN-based methods. Figure 2(a) shows the training curves of selected GNN-based methods, including Decagon, SumGNN, DeepLGF and EmerGNN. Decagon is the most efficient as it only uses a few part of the biomedical network. SumGNN and EmerGNN are slower than Decagon and DeepLGF as they need to learn specific subgraph representations for different drug pairs. However, EmerGNN can achieve higher accuracy than other GNN-based methods in just a few hours, and longer training time has the potential to lead to even better performance. Given that the clinical development of a typical innovative drug usually takes years (Brown et al. 2021), the computation time of EmerGNN is acceptable.

Figure 2(b) shows the GPU memory footprint and the number of parameters of these GNN-based models. It is clear that EmerGNN is memory and parameter efficient. First, its subgraphs for DDI prediction are much smaller than the biomedical

network (as indicated in Figure A.1 in the Supplementary Section A.1). Second, EmerGNN mainly relies on the biomedical concepts instead of the drugs' embeddings to do predictions, resulting in a small number of parameters used. In comparison, DeepLGF requires a large number of model parameters to learn embeddings from the biomedical network.

## 2.6 Analyzing the Learned Paths in the Subgraph

EmerGNN uses attention weights to measure the importance of edges in the subgraph for predicting DDI of the emerging drugs. Here, we analyze what is captured by the attention weights by checking correlations between predicted interaction types with interactions and relations in the path-based subgraphs. Specifically, we first extract top five paths from  $u$  to  $v$  and from  $v$  to  $u$ , respectively, for each triplet  $(u, i_{\text{pred}}, v) \in \mathcal{N}_{\text{D-test}}$  based on the average attention weights of edges in each path with a beam search algorithm. Next, we count how many times an interaction type  $i \in \mathcal{R}_I$  and a relation type  $r \in \mathcal{R}_B$  appears on the selected edges given  $i_{\text{pred}}$  (details of the path extraction algorithm and correlation computation are in Supplementary Section B.2 and B.3).



**Fig. 3** Analyzing relation types on the selected paths on the DrugBank dataset. (a) Correlations between interaction  $i_{\text{pred}}$  of emerging drugs and interaction  $i$  in the augmented network. (b) Comparing the drug pair similarities between two groups of drug pairs. (c) Explanation on the learned correlation between pairs of interaction types in the three yellow circles in (a). (d) Heatmap of biomedical relations. (e) Performance of modified biomedical networks with selected relations.

### Analyzing the drug interaction types

Figure 3(a) shows correlations between interaction  $i_{\text{pred}}$  to be predicted and interaction  $i$  obtained in the subgraphs via path extraction. We can see that the diagonal elements are dominant. This pattern indicates that when predicting a target

interaction  $i_{\text{pred}}$  for  $(u, v)$ , the paths with large attention weights in the subgraph  $\mathcal{G}_{u,v}^L$  are likely to go through another drug (e.g.  $u_1$ ) that has interaction  $i_1 = i_{\text{pred}}$ . We suppose that these drugs like  $u_1$  may have similar properties as the emerging drug  $u$ . To demonstrate this point, we group

the above cases of drug pairs  $(u, u_1)$  as *Group 1* and the other pairs  $(u, u_2)$  with a random drug  $u_2 \in \mathcal{V}_D$  as *Group 2*. We show the histogram distribution of two groups' drug fingerprint similarities of 2000 drug pairs in Figure 3(b). We observe that *Group 1* has a larger quantity of highly similar drug pairs ( $> 0.5$ ) than *Group 2*. This result demonstrates that similar drugs play an important role in predicting the DDIs of emerging drugs and our method can implicitly search for these similar drugs.

Apart from the diagonal part, there exists strongly correlated pairs of interactions. This happens when the emerging drug (e.g.  $u$ ) finds some connections with another drug (e.g.  $u_3$ ) whose intersection  $i_3$  with the existing drug (e.g.  $v$ ) is correlated with  $i_{\text{pred}}$ . We pick up interaction pairs with top-three weights (circled in yellow) and illustrate them in Figure 3(c). For example, we find strong correlation between increasing constipating activity and decreasing analgesic activity, verified by Liu and Wittbrodt (2002).

#### Analyzing the biomedical relation types

We show the correlations between drug interaction  $i_{\text{pred}}$  and biomedical relations  $r \in \mathcal{R}_B$  in Figure 3(d). It can be seen that there are a few relation types consistently selected when predicting different interaction types. In particular, the most frequent relation type is the drug resembling relation CrC. Again, this verifies the importance of similar drugs for emerging drug prediction. The other types frequently selected are related to diseases (CrD), genes (CbG), pharmacologic classes (PCiC) and side effects (CsSE). To analyze the importance of these relations, we compare the performance of EmerGNN with the full biomedical network  $\mathcal{N}_B$  and networks with only top-1 attended relation (CrC, with 0.4% edges), top-3 relations (CrC, CbG, CsSE, with 9.3% edges), or top-5 relations (CrC, CtD, CvG, PCiC, CsSE, with 9.4% edges) in the middle of Figure 3(e). As a comparison, we randomly sample 10%, 30% and 50% edges from  $\mathcal{N}_B$  and show the performance in the right part of Figure 3(e). As shown, keeping the top-1 to 5 relation in  $\mathcal{N}_B$  leads to a comparable performance as a full network  $\mathcal{N}_B$ . However, the performance will significantly deteriorate when edges are randomly dropped. These experiments show that EmerGNN is able to select

important and relevant relations in the biomedical network for DDI prediction.

## 2.7 Case Study on Drug-pairs

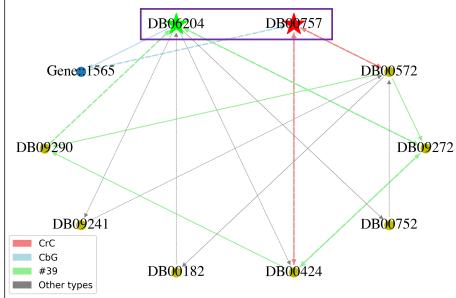
### *Visualization of the learned paths*

We visualize important paths in the subgraph in Figure 4(a) and (b) by selecting top ten paths between  $u$  and  $v$  according to the average of edges' attention weights on each path. We provide a detailed interpretation on one path from  $u$  to  $v$  and the other from  $v$  to  $u$  for each subgraph, with more examples provided in Supplementary Section C.2. Figure 4(a) shows an existing drug DB06204 (Tapentadol) in green star and an emerging drug DB00757 (Dolasetron) in red star, with the predicted relation being “#Drug1 may decrease the analgesic activities of #Drug2” (#52). The first path connects the two drugs firstly through the binding protein Gene::1565 (CYP2D6), which is a P450 enzyme that plays a key role in drug metabolism (Estabrook, 2003). The second path finds a similar drug DB00424 (Hyoscyamine) of DB00757 (Dolasetron) through the resemble relation, and concludes that DB06204 (Tapentadol) may potentially decrease the analgesic activity of DB00757 (Dolasetron) due to the correlation between constipating and analgesic activities as indicated in Figure 3(c). Figure 4(b) shows an emerging drug, DB00598 (Labetalol), and an existing drug, DB00610 (Metaraminol), with a similar explanation for the learned paths. In particular, the second path finds a similar drug DB00421 (Spironolactone), which may decrease the vasoconstricting activity of DB00610 (Metaraminol), providing a hint that Labetalol may also decrease the vasoconstricting activity of Metaraminol. Comparing with the original subgraphs  $\mathcal{G}_{u,v}^L$  with tens of thousands of edges (see Figure A.1 in Supplementary Section A.1), the subgraphs in Figure 4 are much relevant and interpretable to the target prediction. These results show that EmerGNN can find important paths that indicate relevant interaction types and biomedical entities for emerging drug prediction.

### *Visualization of drug pair representations*

Next, we visualize the drug pair representations obtained by CompGCN, SumGNN and EmerGNN. As

(a)



Target: Tapentadol (DB06204) may decrease the analgesic activity of Dolasetron (DB00757).

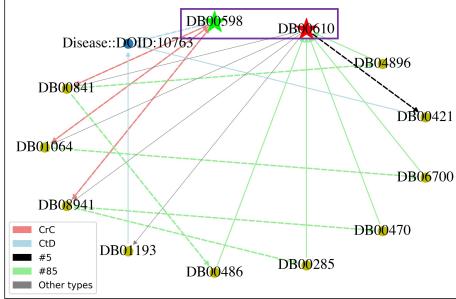
Path1 (0.6666): Tapentadol → CYP2D6 (P450) → Dolasetron

Explanation: Tapentadol can bind the P450 enzyme CYP2D6 (Gene:1565), which is vital for the metabolism of many drugs like Dolasetron (Estabrook, 2003). In addition, Binding of drug to plasma proteins is reversible, and changes in the ratio of bound to unbound drug may lead to drug-drug interactions (Kneip et al. 2008).

Path2 (0.8977): Dolasetron resembles Hyoscymine #39;↑ constipating Eluxadoline #39;<sub>inv</sub> Tapentadol

Explanation: Dolasetron is similar to drug Hyoscymine (DB00424). Hyoscymine and Tapentadol can get some connection since they will both increase the constipating activity of Eluxadoline (DB09272). As suggested by Liu and Wittbrodt (2022), reversing opioid-induced constipation often causes the unwanted side effect of analgesia reversal.

(b)



Target: Labetalol (DB00598) may decrease the vasoconstricting activity of Metaraminol (DB00610).

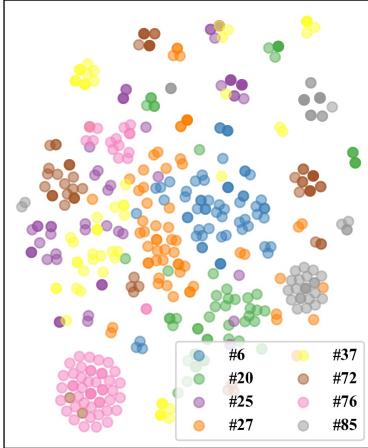
Path1 (0.8274): Labetalol → Isoxsuprime → Dronabinol → Metaraminol

Explanation: Labetalol is similar to the drug Isoxsuprime (DB008941). Isoxsuprime and Metaraminol can get some connection since Dronabinol (DB00470) will increase the tachycardic activity of both of them. As suggested by Chaudhry et al (2022), the decrease of vasoconstriction and the increase of tachycardic are often correlated.

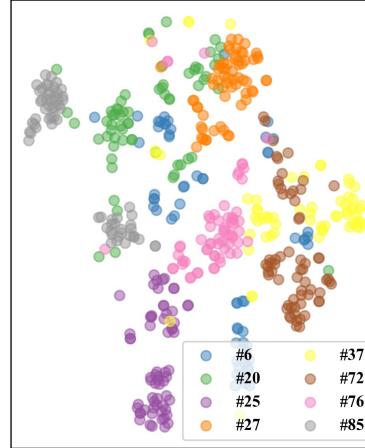
Path2 (0.8175): Metaraminol #5;<sub>inv</sub> vasoconstricting → Spironolactone → hypertension → Labetalol

Explanation: Labetalol and Spironolactone (DB00421) get can some connections since they treat the same disease hypertension (DOID:10763). As Spironolactone may decrease the vasoconstricting activity of Metaraminol (indicated by the inverse edge), we predict that Labetalol may also decrease the vasoconstricting activity of Metaraminol.

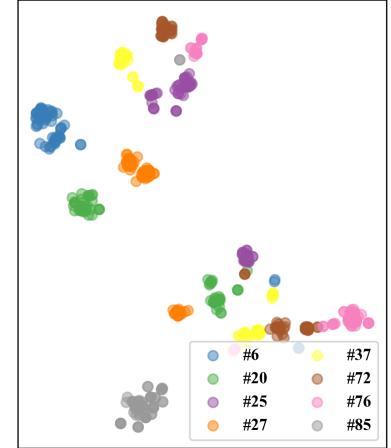
(c)



(d)



(e)



**Fig. 4** Visualization of drug pairs. (a-b) The learned important paths. The drug pairs to be predicted are highlighted in purple rectangles; dashed lines mean reverse types; “other types” in gray edges mean the interaction types aside from the given ones. (c-e) t-SNE of concatenated representations in CompGCN, subgraph representations in SumGNN, and pair-wise representations  $h_{u,v}^{(L)}$  in EmerGNN. The legends in the figures specify the ID of the interaction type; each dot denotes an interaction  $(u, i, v)$ ; the different colors in dots indicate the interactions that the drug pairs  $(u, v)$  have.

CompGCN embeds each entity separately, we concatenate embeddings of the two drugs’ representations for a given drug pair  $(u, v)$ . SumGNN encodes the enclosing subgraphs from  $u$  and  $v$  for interaction prediction, thus we take the representation of the interacted enclosing subgraph as the drug pair representation. Finally, the drug pair representation is directly given by  $h_{u,v}^{(L)}$  for EmerGNN. Since there are too many interaction types and

drug pairs in  $\mathcal{N}_{\text{D-test}}$ , we randomly sample eight interaction types and 64 drug pairs for each interaction type. The visualization results obtained by t-SNE (Van der Maaten and Hinton, 2008) for above methods are in Figure 4(c)-(e). As shown, the drug pairs with the same interaction are more densely gathered in EmerGNN than CompGCN and SumGNN. This means that the drug pair representations of EmerGNN can better separate the different

interaction types. As a result, **EmerGNN** is able to learn better drug pair representations than the other GNN methods, **CompGCN** and **SumGNN**.

## 2.8 Ablation Studies

In this part, we conduct ablation studies on the DrugBank dataset to analyze the strengths of proposed method in detail.

### *Interaction-wise performance*

In Figure 5(a), we compare the performance of top-performing models according to the frequency of interaction types to deeply analyze the different models' ability. Specifically, we group the interaction types into five groups based on their frequency in the dataset and show the average macro F1 performance in each group. **EmerGNN** outperforms the baselines in all frequencies. For the high frequency relations (1%~20%), all the methods, except for **KG-DDI**, have good performance. For extremely low frequency relations (81%~100%), all the methods work poorly. The performance of all methods deteriorates in general for relations with a lower frequency. However, the relative performance gain of **EmerGNN** tends to be larger, especially in the range of 61%~80%. These results indicate **EmerGNN**'s strengths in generalization and ability to extract essential information from the biomedical network for predicting rare drugs and interaction types.

### *Adding interactions to emerging drugs*

The experiments in Section 2.4 study the scenario of emerging drugs without any interaction to existing drugs. In practice, we may have a few known interactions between the emerging and existing drugs (often obtained from limited clinical trials). Hence, we analyze how different models perform if we add a few interactions between the emerging drugs and the existing drugs. This is conducted by randomly sampling 1/3/5 interaction edges in  $\mathcal{N}_{D\text{-test}}$  for each emerging drug in  $\mathcal{V}_{D\text{-test}}$ , and move these edges to  $\mathcal{N}_{D\text{-train}}$ . We select some representative baselines and show the F1-scores in Figure 5(b). We can see that the shallow models that can be generally used for emerging drug prediction, i.e., **CSMDDI** and **HIN-DNN**, do not change much since the features they use are unchanged. The methods learning drug embeddings, such as **KG-DDI** and **DeepLGF**, enjoy more significant

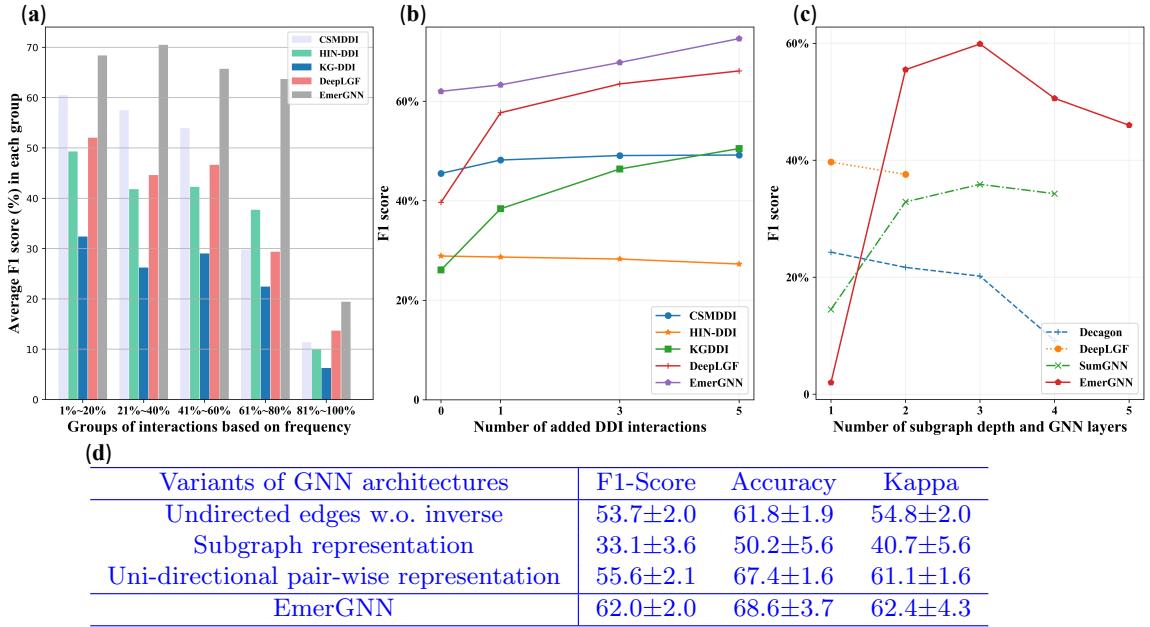
improvement when additional knowledge is provided. In comparison, **EmerGNN** has increased performance with more interactions added and is still the best over all the compared methods.

### *Changing the length $L$*

The value of  $L$  determines the maximum number of hops of neighboring entities that the GNN-based models can visit. We compare the F1-score of different values of  $L$ , i.e., the number of GNN layers in **Decagon** and **DeepLGF**, the depth of enclosing subgraph in **SumGNN**, and the depth of path-based subgraph in **EmerGNN**, in Figure 5(c). The performance of **Decagon** and **DeepLGF** gets worse when  $L$  gets larger. Considering that **Decagon** and **DeepLGF** work on the full biomedical network, too many irrelevant information will be involved in the representation learning, leading to worse performance. **DeepLGF** runs out-of-memory when  $L \geq 3$ . For **SumGNN** and **EmerGNN**,  $L = 1$  performs the worst as the information is hard to be passed from the emerging drug to the existing drug. Since **SumGNN** can leverage the drug features for prediction, it outperforms **Decagon**. In comparison, when  $L$  increases from 1 to 3, **EmerGNN** can benefit much from the relevant information on the biomedical network. However, the performance of **EmerGNN** will decrease when  $L > 3$ . Intuitively, the path-based subgraph will contain too much irrelevant information when the length gets longer, increasing the learning difficulty. Hence, a moderate number of path length, i.e.,  $L = 3$ , is optimal for **EmerGNN**, considering both the effectiveness and computation efficiency.

### *GNN architecture design*

We conduct experiments to analyze the influence of GNN architecture design for the path-based subgraph  $\mathcal{G}_{u,v}^L$  in **EmerGNN** in Figure 5(d). First, we design a simple variant, denoted as Subgraph representation, which learns a subgraph representation as **SumGNN** upon  $\mathcal{G}_{u,v}^L$ . Second, we provide a variant with uni-directional pair-wise representation  $\mathbf{h}_{u,v}^{(L)}$ , which only learns on the path-based subgraph  $\mathcal{G}_{u,v}^L$  from  $u$  to  $v$ . Third, we provide a variant by using undirected edges instead of adding inverse edges in the augmented network. The performance of **EmerGNN**, which learns bi-directional pair-wise representations  $[\mathbf{h}_{u,v}^{(L)}, \mathbf{h}_{v,u}^{(L)}]$  on both  $\mathcal{G}_{u,v}^L$  from  $u$  to  $v$  and  $\mathcal{G}_{v,u}^L$  from  $v$  to  $u$



**Fig. 5** Ablation studies on the DrugBank dataset. (a) Performance comparison of interaction groups based on interaction frequency. (b) Performance comparison of adding interaction edges for emerging drugs into the training set  $\mathcal{N}_{D\text{-train}}$ . (c) Performance comparison of GNN-based methods by varying the number of layers. (d) Performance comparison of different GNN architecture designs.

(see more details in Section 4.1), is added in the bottom row as a reference.

Using undirected edges without introducing the inverse edge will influence the effectiveness of **EmerGNN** as the directional information on the biomedical network is lost. Comparing subgraph representation with uni-directional pair-wise representation, we observe that the flow-based GNN architecture is more effective than the GNN used in **SumGNN**. Even though uni-directional pair-wise representation can achieve better performance compared with all the baselines in the top part of Table 1, learning bi-directional representations can help to further improve the prediction ability by balancing the bi-directional communications between drugs.

### 3 Conclusion

Drug-drug interactions (DDIs) can modulate the safety and efficacy of medications by altering their pharmacology and pharmacokinetics. Predicting DDIs using deep learning methods is a pressing challenge for drug discovery and development. In this paper, we focus on predicting DDIs in a cold-start scenario, where at least one of the drug is

new and has no known interaction with existing drugs. This scenario is more realistic and challenging than the conventional DDI prediction task, where both drugs are well-established and have abundant interaction data. Emerging drugs are often developed to address unmet medical needs, such as new diseases, drug-resistant infections, rare conditions, and ineffective/unsafe treatments. Accurately predicting interactions for emerging drugs can help health care providers make more informed decisions about medication management and avoid adverse drug reactions. Existing deep learning methods do not perform well in this scenario, as they require large amounts of data to train their over-parameterized models.

We propose **EmerGNN**, a graph neural network (GNN)-based method that can predict DDIs between emerging and existing drugs. **EmerGNN** learns pair-wise representations of drugs by integrating the biomedical entities and relations that connect them. To effectively extract essential information from the biomedical entities and relations, we customize a flow-based GNN architecture that propagates and transforms the drug pair representation from the emerging drug to the existing drug. Moreover, the attention weights on

the edges highlight the important paths between the existing and emerging drugs, which provide interpretable evidence for the prediction. Extensive experiments demonstrate that our method outperforms existing deep learning approaches by a large margin. The significant improvement on accuracy shows the potential of EmerGNN in real clinical practice to improve patient care or contribute to more efficient drug development processes.

Although we demonstrate the effectiveness of EmerGNN for DDI prediction in this paper, EmerGNN is a general approach that can be applied to other biomedical applications, such as protein-protein interaction prediction, drug-target interaction prediction and disease-gene interaction prediction. We anticipate that the paths attended by EmerGNN can enhance the accuracy and interpretability of these predictions. We hope that our open-sourced EmerGNN can serve as a novel algorithm and a unique deep learning tool to advance biomedicine and healthcare, by enabling practitioners to exploit the rich knowledge in existing large biomedical networks for low-data scenarios.

## 4 Methods

### 4.1 Architecture of EmerGNN

To predict interactions between emerging drugs and existing drugs, it is important to leverage relevant information in the biomedical network. Our framework contains four main components: (i) constructing an augmented network by integrating the DDI network with the biomedical network and adding inverse edges; (ii) extracting all the paths with length no longer than  $L$  from  $u$  to  $v$  to construct a path-based subgraph  $\mathcal{G}_{u,v}^L$ ; (iii) encoding pair-wise subgraph representation  $\mathbf{h}_{u,v}^{(L)}$  by a flow-based GNN with attention mechanism such that the information can flow from  $u$  over the important entities and edges in  $\mathcal{G}_{u,v}^L$  to  $v$ ; (iv) predicting the interaction type based on the bi-directional pair-wise subgraph representations. The overall framework is shown in Figure 1.

### Augmented network

Given the DDI network  $\mathcal{N}_D = \{(u, i, v) : u, v \in \mathcal{V}_D, i \in \mathcal{R}_I\}$ <sup>3</sup> and the biomedical network  $\mathcal{N}_B = \{(h, r, t) : h, t \in \mathcal{V}_B, r \in \mathcal{R}_B\}$ , we integrate the two networks into

$$\mathcal{N}' = \mathcal{N}_D \cup \mathcal{N}_B = \{(e, r, e') : e, e' \in \mathcal{V}', r \in \mathcal{R}'\},$$

with  $\mathcal{V}' = \mathcal{V}_D \cup \mathcal{V}_B$  and  $\mathcal{R}' = \mathcal{R}_I \cup \mathcal{R}_B$ . The integrated network  $\mathcal{N}'$  connects the existing and emerging drugs by concepts in the biomedical network. Since the relation types are directed, we follow the common practices in knowledge graph learning (Vashishth et al., 2019; Yu et al., 2021) to add inverse types. Specifically, we add  $r_{inv}$  for each  $r \in \mathcal{R}'$  and create a set of inverse types  $\mathcal{R}'_{inv}$ , which subsequently leads to an inverse network

$$\mathcal{N}'_{inv} = \{(e', r_{inv}, e) : (e, r, e') \in \mathcal{N}'\}.$$

Note that the inverse relations will not influence the information in the original biomedical network since we can transform any inverse edge  $(e', r_{inv}, e)$  back to the original edge  $(e, r, e')$ . Semantically, the inverse relations can be regarded as a kind of active voice vs. passive voice in linguistics, e.g., *includes\_inv* can be regarded as *being included* and *causes\_inv* can be regarded as *being caused*. By adding the inverse edges, the paths can be smoothly organized in single directions. For example, a path  $a \xrightarrow{r_1} b \xleftarrow{r_2} c$  can be transformed to  $a \xrightarrow{r_1} b \xrightarrow{r_2, inv} c$ , which is more computational friendly.

After the above two steps, we obtain the augmented network

$$\mathcal{N} = \mathcal{N}' \cup \mathcal{N}'_{inv} = \{(e, r, e') : e, e' \in \mathcal{V}, r \in \mathcal{R}\},$$

with entity set  $\mathcal{V} = \mathcal{V}' = \mathcal{V}_D \cup \mathcal{V}_B$  and relation set  $\mathcal{R} = \mathcal{R}' \cup \mathcal{R}'_{inv}$ .

### Path-based subgraph formulation

Inspired by the path-based methods in knowledge graph learning (Lao et al., 2011; Xiong et al., 2017), we are motivated to extract the paths connecting existing and emerging drugs, and predict the interaction type based on the paths.

---

<sup>3</sup> $\mathcal{N}_D$  is specified as  $\mathcal{N}_{D\text{-train}}/\mathcal{N}_{D\text{-valid}}/\mathcal{N}_{D\text{-test}}$  in the training/validation/testing stages, respectively, so does  $\mathcal{N}_B$ .

Given a drug pair  $(u, v)$  to be predicted, we extract the set  $\mathcal{P}_{u,v}^L$  of all the paths with length up to  $L$ . Each path in  $\mathcal{P}_{u,v}^L$  has the form

$$e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_L} e_L,$$

with  $e_0 = u$ ,  $e_L = v$  and  $(e_{i-1}, r_i, e_i) \in \mathcal{N}$ ,  $i = 1, \dots, L$ . The intermediate entities  $e_1, \dots, e_{L-1} \in \mathcal{V}$  can be drugs, genes, diseases, side-effects, symptoms, pharmacologic class, etc., and  $r_1, \dots, r_L \in \mathcal{R}$  are the interactions or relations between the biomedical entities. In order to preserve the local structures, we merge the paths in  $\mathcal{P}_{u,v}^L$  to a subgraph  $\mathcal{G}_{u,v}^L$  such that the same entities are merged to a single node. The detailed steps of path extraction and subgraph generation are provided in Supplementary material B.1.

Different from the subgraph structures used for link prediction on general graphs (Zhang and Chen, 2018; Teru et al, 2020; Yu et al, 2021), the edges in  $\mathcal{G}_{u,v}^L$  are pointed away from  $u$  and towards  $v$ . Our objective is to learn a GNN  $g(\cdot)$  with parameters  $\theta$  that predicts DDI between  $u$  and  $v$  based on the path-based subgraph  $\mathcal{G}_{u,v}^L$ , i.e.,

$$\text{DDI}(u, v) = g(\mathcal{G}_{u,v}^L; \theta). \quad (1)$$

The link prediction problem on the DDI network is then transformed as a whole graph learning problem.

### Flow-based GNN architecture

Given  $\mathcal{G}_{u,v}^L$ , we would like to integrate essential information in it to predict the target interaction type. Note that the edges in  $\mathcal{G}_{u,v}^L$  are from the paths  $\mathcal{P}_{u,v}^L$  connecting from  $u$  to  $v$ . We aim to design a special GNN architecture that the information can flow from drug  $u$  to  $v$ , via integrating entities and relations in  $\mathcal{G}_{u,v}^L$ .

Denote  $\mathcal{V}_{u,v}^\ell, \ell = 0, \dots, L$ , as the set of entities that can be visited in the  $\ell$ -th flow step from  $u$  (like the four ellipses in  $g(\mathcal{G}_{u,v}^L; \theta)$  in Figure 1). In particular, we have  $\mathcal{V}_{u,v}^0 = \{u\}$  as the starting point and  $\mathcal{V}_{u,v}^L = \{v\}$  as the ending point. In the  $\ell$ -th iteration, the visible entities in  $\mathcal{V}_{u,v}^\ell$  contains entities that are  $\ell$ -steps away from drug  $u$  and are  $(L - \ell)$ -steps away from drug  $v$  in the augmented network  $\mathcal{N}$ . We use the fingerprint features (Rogers and Hahn, 2010) of drug  $u$  as the input representation of  $u$ , namely  $\mathbf{h}_{u,u}^{(0)} = \mathbf{f}_u$ .

Then, we conduct message flow for  $L$  steps with the function

$$\mathbf{h}_{u,e}^{(\ell)} = \delta \left( \mathbf{W}^{(\ell)} \sum_{e' \in \mathcal{V}_{u,v}^{\ell-1}} (\mathbf{h}_{u,e'}^{(\ell-1)} + \phi(\mathbf{h}_{u,e'}^{(\ell-1)}, \mathbf{h}_r^{(\ell)})) \right), \quad (2)$$

for entities  $e \in \mathcal{V}_{u,v}^\ell$ , where  $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d \times d}$  is a learnable weighting matrix for step  $\ell$ ;  $\mathbf{h}_{u,e'}^{(\ell-1)}$  is the pair-wise representation of entity  $e' \in \mathcal{V}_{u,v}^{\ell-1}$ ;  $r$  is the relation type between  $e'$  and  $e$ ;  $\mathbf{h}_r^{(\ell)} \in \mathbb{R}^d$  is the learnable representation with dimension  $d$  of  $r$  in the  $\ell$ -th step; and  $\phi(\cdot, \cdot) : (\mathbb{R}^d, \mathbb{R}^d) \rightarrow \mathbb{R}^d$  is the function combining the two vectors; and  $\delta(\cdot)$  is the activation function ReLU (Nair and Hinton, 2010).

Since the biomedical network is not specially designed for the DDI prediction task, we need to control the importance of different edges in  $\mathcal{G}_{u,v}^L$ . We use a drug-dependent attention weight for function  $\phi(\cdot, \cdot)$ . Specifically, we design the message function for each edge  $(e', r, e)$  during the  $l$ -th propagation step as

$$\phi(\mathbf{h}_{u,e'}^{(\ell-1)}, \mathbf{h}_r^{(\ell)}) = \alpha_r^{(\ell)} \cdot (\mathbf{h}_{u,e'}^{(\ell-1)} \odot \mathbf{h}_r^{(\ell)}), \quad (3)$$

where  $\odot$  is an element-wise dot product of vectors and  $\alpha_r^{(\ell)}$  is the attention weight controlling the importance of messages. We design the attention weight depending on the edges' relation type as

$$\alpha_r^{(\ell)} = \sigma \left( (\mathbf{w}_r^{(\ell)})^\top [\mathbf{f}_u; \mathbf{f}_v] \right),$$

where the relation weight  $\mathbf{w}_r^{(\ell)} \in \mathbb{R}^{2d}$  is multiplied with the fingerprints  $[\mathbf{f}_u; \mathbf{f}_v] \in \mathbb{R}^{2d}$  of drugs to be predicted and  $\sigma(\cdot)$  is a sigmoid function returning a value in  $(0, 1)$ .

After iterating for  $L$  steps, we can obtain the representation  $\mathbf{h}_{u,v}^{(L)}$  that encodes the important paths up to length  $L$  between drugs  $u$  and  $v$ .

### Objective and training

In practice, the interaction types can be symmetric, e.g., #Drug1 and #Drug2 may have the side effect of headache if used together, or asymmetric, e.g., #Drug1 may decrease the analgesic activities of #Drug2. Besides, the emerging drug can appear in either the source (drug  $u$ ) or target (drug  $v$ ). We extract the reverse subgraph  $\mathcal{G}_{v,u}^L$  and encode it with the same parameters in (2) to obtain the

reverse pair-wise representation  $\mathbf{h}_{v,u}^{(L)}$ . Then the bi-directional representations are concatenated to predict the interaction type with

$$\mathbf{l}(u, v) = \mathbf{W}_{\text{rel}} \left[ \mathbf{h}_{u,v}^{(L)}; \mathbf{h}_{v,u}^{(L)} \right]. \quad (4)$$

Here, the transformation matrix  $\mathbf{W}_{\text{rel}} \in \mathbb{R}^{|\mathcal{R}_1| \times 2d}$  is used to map the pair-wise representations into prediction logits  $\mathbf{l}(u, v)$  of the  $|\mathcal{R}_1|$  interaction types. The  $i$ -th logit  $l_i(u, v)$  indicates the plausibility of interaction type  $i$  being predicted. The full algorithm and implementation details of (4) are provided in Supplementary Section B.1.

Since we have two kinds of tasks, i.e., multi-class (on the DrugBank dataset) and multi-label (on the TWOSIDES dataset) interaction predictions, the training objectives are different.

For DrugBank, there exists at most one interaction type between two drugs. Given two drugs  $u$  and  $v$ , once we obtain the prediction logits  $\mathbf{l}(u, v)$  of different interaction types, we use a softmax function to compute the probability of each interaction type, i.e.,

$$I_i(u, v) = \frac{\exp(l_i(u, v))}{\sum_{j \in \mathcal{R}_1} \exp(l_j(u, v))}.$$

Denote  $\mathbf{y}(u, v) \in \mathbb{R}^{|\mathcal{R}_1|}$  as the ground-truth indicator of target interaction type, where  $y_i(u, v) = 1$  if  $(u, i, v) \in \mathcal{N}_D$ , otherwise zero. We minimize the following cross-entropy loss to train the model parameters

$$\mathcal{L}_{\text{DB}} = - \sum_{(u, i, v) \in \mathcal{N}_{\text{D-train}}} y_i(u, v) \log I_i(u, v). \quad (5)$$

For TWOSIDES, there may be multiple interactions between two drugs. The objective is to predict whether there is an interaction  $p$  between two drugs. Given two drugs  $u, v$  and the prediction logits  $\mathbf{l}(u, v)$ , we use the sigmoid function

$$I_i(u, v) = \frac{1}{1 + \exp(-l_i(u, v))},$$

to compute the probability of interaction type  $i$ . Different with the multi-class task in DrugBank,

we use the binary cross entropy loss

$$\begin{aligned} \mathcal{L}_{\text{TS}} = & - \sum_{(u, i, v) \in \mathcal{N}_{\text{D-train}}} \left( \log(I_i(u, v)) \right. \\ & \left. + \sum_{(u', v') \in \mathcal{N}_i} \log(1 - I_i(u', v')) \right), \end{aligned} \quad (6)$$

where  $\mathcal{N}_i$  is the set of drug pairs that do not have the interaction type  $i$ . Details of  $\mathcal{N}_i$  are provided in Supplementary Section A.1.

We use stochastic gradient optimizer Adam (Kingma and Ba, 2014) to optimize the model parameters

$$\boldsymbol{\theta} = \left\{ \mathbf{W}_{\text{rel}}, \{ \mathbf{W}^{(\ell)}, \mathbf{h}_r^{(\ell)}, \mathbf{w}_r^{(\ell)} \}_{\ell=1, \dots, L, r \in \mathcal{R}} \right\},$$

by minimizing loss function (5) for the DrugBank dataset or loss (6) for the TWOSIDES dataset.

## 4.2 Link Prediction with Deep Networks

The general pipeline for GNN-based link prediction contains three parts: subgraph extraction, node labeling, and GNN learning. Take SEAL (Zhang and Chen, 2018) as an example. It firstly extracts enclosing subgraph, which contains the intersection of  $L$ -hop neighbors of  $(u, v)$ , between drug pairs. In order to distinguish the nodes on the subgraph, SEAL then labels the nodes based on their shortest path distance to both nodes  $u$  and  $v$ . Finally, a GNN aggregates the representations of node labels for  $L$  steps, integrates the representations of all the nodes, and predicts the interaction based on the integrated representation.

On subgraph extraction, we use union of paths to form a path-based subgraph instead of the enclosing subgraph, as we need to integrate the entities and relations on the augmented network while propagating from drug  $u$  to  $v$ . Node labeling is difficult to be extended to heterogeneous graph, and a simple extension from homogeneous graph may not lead to good performance (Teru et al, 2020). In comparison, our designed flow-based GNN avoids the labeling problem by propagating from  $u$  to  $v$  step-by-step. Benefiting by the propagation manner, we do not need use an extra pooling layer (Zhang and Chen, 2018; Vashisht et al, 2019; Teru et al, 2020) and just use the

pair-wise representation  $\mathbf{h}_{u,v}^{(L)}$  in the last step to encode  $\mathcal{G}_{u,v}^L$ . The above benefits are demonstrated in Figure 5(c) and we provide more analysis in Supplementary Section B.4.

## 5 Data and Code Availability

Our data and code are available at <https://github.com/yzhangee/EmerGNN>.

## References

- Bergstra J, Komer B, Eliasmith C, et al (2015) Hyperopt: A Python library for model selection and hyperparameter optimization. *Computational Science & Discovery* 8(1):014,008
- Brown DG, Wobst HJ, Kapoor A, et al (2021) Clinical development times for innovative drugs. *Nature reviews Drug discovery* 21(11):793–794
- Cohen J (1960) A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1):37–46
- Dewulf P, Stock M, De Baets B (2021) Cold-start problems in data-driven prediction of drug-drug interaction effects. *Pharmaceuticals* 14(5):429
- Estabrook RW (2003) A passion for P450s (remembrances of the early history of research on cytochrome P450). *Drug Metabolism and Disposition* 31(12):1461–1473
- Gilmer J, Schoenholz SS, Riley PF, et al (2017) Neural message passing for quantum chemistry. In: International Conference on Machine Learning, PMLR, pp 1263–1272
- Himmelstein DS, Lizze A, Hessler C, et al (2017) Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife* 6:e26,726
- Huang K, Xiao C, Glass LM, et al (2020) SkipGNN: predicting molecular interactions with skip-graph networks. *Scientific Reports* 10(1):1–16
- Jiang H, Lin Y, Ren W, et al (2022) Adverse drug reactions and correlations with drug-drug interactions: A retrospective study of reports from 2011 to 2020. *Frontiers in Pharmacology* 13
- Karim MR, Cochez M, Jares JB, et al (2019) Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-LSTM network. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp 113–123
- Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. Tech. rep., arXiv:1412.6980
- Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations
- Lao N, Mitchell T, Cohen W (2011) Random walk inference and learning in a large scale knowledge base. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp 529–539
- Ledford H (2022) Hundreds of COVID trials could provide a deluge of new drugs. *Nature* pp 25–27
- Letinier L, Cossin S, Mansiaux Y, et al (2019) Risk of drug-drug interactions in out-hospital drug dispensings in France: Results from the drug-drug interaction prevalence study. *Frontiers in Pharmacology* 10:265
- Lin X, Quan Z, Wang ZJ, et al (2020) KGNN: Knowledge graph neural network for drug-drug interaction prediction. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp 2739–2745
- Liu M, Wittbrodt E (2002) Low-dose oral naloxone reverses opioid-induced constipation and analgesia. *Journal of Pain and Symptom Management* 23(1):48–53
- Liu Z, Wang XN, Yu H, et al (2022) Predict multi-type drug-drug interactions in cold start scenario. *BMC Bioinformatics* 23(1):75
- Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *Journal of Machine Learning Research* 9(11)

- Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning, pp 807–814
- Percha B, Altman RB (2013) Informatics confronts drug-drug interactions. *Trends in Pharmacological Sciences* 34(3):178–184
- Ren ZH, You ZH, Yu CQ, et al (2022) A biomedical knowledge graph-based method for drug-drug interactions prediction through combining local and global features with deep neural networks. *Briefings in Bioinformatics* 23(5):bbac363
- Rogers D, Hahn M (2010) Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* 50(5):742–754
- Su X, Wang H, Zhao N, et al (2022) Trends in innovative drug development in China. *Nature Reviews Drug Discovery*
- Tanvir F, Islam MIK, Akbas E (2021) Predicting drug-drug interactions using meta-path based similarities. In: IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, IEEE, pp 1–8
- Tatonetti NP, Ye PP, Daneshjou R, et al (2012) Data-driven prediction of drug effects and interactions. *Science translational medicine* 4(125):125ra31–125ra31
- Teru K, Denis E, Hamilton W (2020) Inductive relation prediction by subgraph reasoning. In: International Conference on Machine Learning, PMLR, pp 9448–9457
- Vashisht S, Sanyal S, Nitin V, et al (2019) Composition-based multi-relational graph convolutional networks. In: International Conference on Learning Representations
- Vilar S, Uriarte E, Santana L, et al (2014) Similarity-based modeling in large-scale prediction of drug-drug interactions. *Nature Protocols* 9(9):2147–2163
- Wishart DS, Feunang YD, Guo AC, et al (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research* 46(D1):D1074–D1082
- Xiong W, Hoang T, Wang WY (2017) DeepPath: A reinforcement learning method for knowledge graph reasoning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 564–573
- Xu K, Hu W, Leskovec J, et al (2018) How powerful are graph neural networks? In: International Conference on Learning Representations
- Yao J, Sun W, Jian Z, et al (2022) Effective knowledge graph embeddings based on multidirectional semantics relations for polypharmacy side effects prediction. *Bioinformatics* 38(8):2315–2322
- Yu H, Zhao S, Shi J (2022) STNN-DDI: a substructure-aware tensor neural network to predict drug-drug interactions. *Briefings in Bioinformatics* 23(4):bbac209
- Yu Y, Huang K, Zhang C, et al (2021) SumGNN: multi-typed drug interaction prediction via efficient knowledge graph summarization. *Bioinformatics* 37(18):2988–2995
- Zhang M, Chen Y (2018) Link prediction based on graph neural networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp 5171–5181
- Zitnik M, Agrawal M, Leskovec J (2018) Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34(13):i457–i466

## A Implementation details

### A.1 Datasets

**Drug-drug interaction.** Following (Zitnik et al, 2018; Yu et al, 2021), we use two benchmark datasets, DrugBank (Wishart et al, 2018) and TWOSIDES (Tatonetti et al, 2012), as the interaction network. The statistics of the two datasets are provided in Table A.1.

**Table A.1** Statistics of common DDI datasets used for predicting interactions between existing drugs.

Statistics	$ \mathcal{V}_D $	$ \mathcal{R}_I $	$ \mathcal{N}_{D\text{-train}} $	$ \mathcal{N}_{D\text{-valid}} $	$ \mathcal{N}_{D\text{-test}} $
Drugbank	1,710	86	134,641	19,224	38,419
TWOSIDES	604	200	177,568	24,887	49,656

When predicting DDIs for emerging drugs, i.e., the S1 and S2 settings, we randomly split  $\mathcal{V}_D$  into three disjoint sets, i.e.,  $\mathcal{V}_D = \mathcal{V}_{D\text{-train}} \cup \mathcal{V}_{D\text{-valid}} \cup \mathcal{V}_{D\text{-test}}$  and  $\mathcal{V}_{D\text{-train}} \cap \mathcal{V}_{D\text{-valid}} \cap \mathcal{V}_{D\text{-test}} = \emptyset$ , where  $\mathcal{V}_{D\text{-train}}$  is the set of existing drugs used for training,  $\mathcal{V}_{D\text{-valid}}$  is the set of emerging drugs for validation, and  $\mathcal{V}_{D\text{-test}}$  is the set of emerging drugs for testing.

The interaction network for training is defined as  $\mathcal{N}_{D\text{-train}} = \{(u, i, v) \in \mathcal{N}_D : u, v \in \mathcal{V}_{D\text{-train}}\}$ . In the S1 setting, we set

- $\mathcal{N}_{D\text{-valid}} = \{(u, i, v) \in \mathcal{N}_D : u \in \mathcal{V}_{D\text{-train}}, v \in \mathcal{V}_{D\text{-valid}}\} \cup \{(u, i, v) \in \mathcal{N}_D : u \in \mathcal{V}_{D\text{-valid}}, v \in \mathcal{V}_{D\text{-train}}\}$ ;
- $\mathcal{N}_{D\text{-test}} = \{(u, i, v) \in \mathcal{N}_D : u \in (\mathcal{V}_{D\text{-train}} \cup \mathcal{V}_{D\text{-valid}}), v \in \mathcal{V}_{D\text{-test}}\} \cup \{(u, i, v) \in \mathcal{N}_D : u \in \mathcal{D}_{D\text{-test}}, v \in (\mathcal{V}_{D\text{-train}} \cup \mathcal{V}_{D\text{-valid}})\}$ .

In the S2 setting, we set  $\mathcal{N}_{D\text{-valid}} = \{(u, i, v) \in \mathcal{N}_D : u, v \in \mathcal{V}_{D\text{-valid}}\}$  and  $\mathcal{N}_{D\text{-test}} = \{(u, i, v) \in \mathcal{N}_D : u, v \in \mathcal{V}_{D\text{-test}}\}$ .

For the TWOSIDES dataset, we follow Yu et al (2021) to randomly sample one negative sample for each  $(u, i, v) \in \mathcal{N}_{D\text{-valid}} \cup \mathcal{N}_{D\text{-test}}$  to form the negative set  $\mathcal{N}_i$  in the evaluation phase. Specifically, if  $u$  is an emerging drug, we randomly sample an existing drug  $v' \in \mathcal{V}_{D\text{-train}}$  and make sure that the new interaction does not exist, i.e.,  $(u, i, v') \notin \mathcal{N}_D$ ; if  $v$  is an emerging drug, we randomly sample an existing drug  $u' \in \mathcal{V}_{D\text{-train}}$  and make sure that the new interaction does not exist, i.e.,  $(u', i, v) \notin \mathcal{N}_D$ .

The detailed statistics of the customized datasets for emerging drug prediction are provided in Table A.2.

**Table A.2** Statistics of datasets used for predicting interactions for emerging drugs in the S1 and S2 settings.

Data	seed	$ \mathcal{V}_{D\text{-train}} $	$ \mathcal{V}_{D\text{-valid}} $	$ \mathcal{V}_{D\text{-test}} $	$ \mathcal{R}_I $	$ \mathcal{N}_{D\text{-train}} $	S1		S2	
							$ \mathcal{N}_{D\text{-valid}} $	$ \mathcal{N}_{D\text{-test}} $	$ \mathcal{N}_{D\text{-valid}} $	$ \mathcal{N}_{D\text{-test}} $
DrugBank	1	1,461	79	161	86	137,864	17,591	32,322	536	1,901
	12	1,465	79	161	86	140,085	17,403	30,731	522	1,609
	123	1,466	81	161	86	140,353	14,933	32,845	396	1,964
	1234	1,463	81	162	86	139,141	15,635	33,254	434	1,956
	12345	1,461	80	169	86	133,394	17,784	35,803	546	2,355
TWOSIDES	1	514	30	60	200	185,673	16,113	45,365	467	2,466
	12	514	30	60	200	172,351	23,815	48,638	717	3,373
	123	514	30	60	200	181,257	18,209	46,969	358	2,977
	1234	514	30	60	200	186,104	25,830	35,302	837	1,605
	12345	514	30	60	200	179,993	22,059	43,867	702	2,695

**Biomedical knowledge graph.** In this work, same as the DDI network, we use different variants of KGs for training, validation and testing. The well-built biomedical KG HetioNet (Himmelstein et al, 2017)

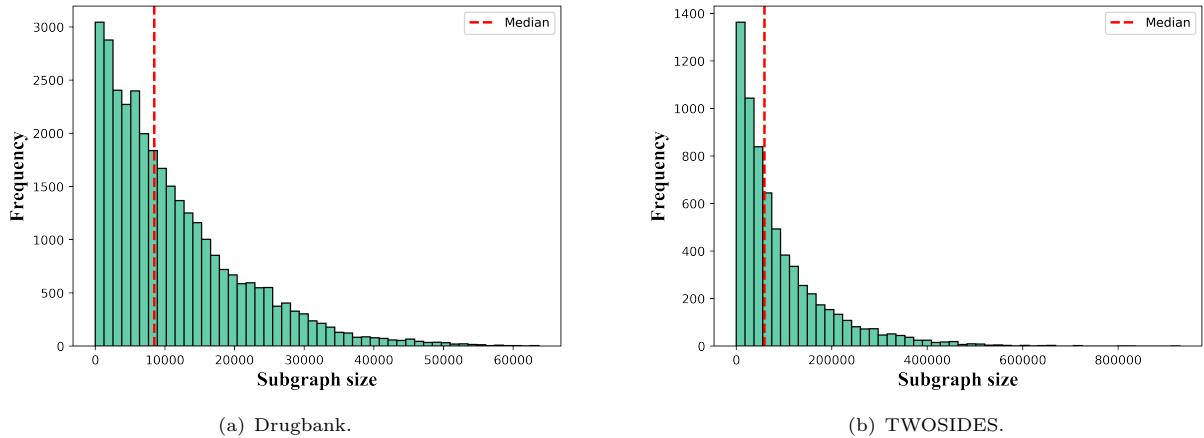
is used here. Denote  $\mathcal{V}_B, \mathcal{R}_B, \mathcal{N}_B$  as the set of entities, relations and edges, respectively, in the original biomedical network. When predicting interactions between existing drugs in the general setting, all the edges in  $\mathcal{N}_B$  are used for training, validation and testing. When predicting interactions between emerging drugs and existing drugs, we use different parts of the biomedical networks.

In order to guarantee that the emerging drugs are connected with some existing drugs through the biomedical entities, we constrain the split of drugs to satisfy the conditions  $\mathcal{V}_{D\text{-valid}} \subset \mathcal{V}_B$  and  $\mathcal{V}_{D\text{-test}} \subset \mathcal{V}_B$ . Meanwhile, we also guarantee that the emerging drugs will not be seen in the biomedical network during training. To achieve this goal, the edges for training are in the set  $\mathcal{N}_{B\text{-train}} = \{(h, r, t) \in \mathcal{N}_B : h, t \notin (\mathcal{V}_{D\text{-valid}} \cup \mathcal{V}_{D\text{-test}})\}$ ; the edges for validation are in the set  $\mathcal{N}_{B\text{-valid}} = \{(h, r, t) \in \mathcal{N}_B : h, t \notin \mathcal{V}_{D\text{-test}}\}$ ; and the testing network is the original network, i.e.,  $\mathcal{N}_{B\text{-test}} = \mathcal{N}_B$ .

**Table A.3** Statistics of the biomedical network used for DDI prediction.

Statistics	$ \mathcal{V}_B $	$ \mathcal{R}_B $	$\mathcal{N}_B$	$ \mathcal{N}_{B\text{-train}} $	$ \mathcal{N}_{B\text{-valid}} $	$ \mathcal{N}_{B\text{-test}} $
HetioNet	34,124	23	1,690,693	1,671,617	1,678,528	1,690,693

In addition, we plot the size distribution (measured by the number of edges in  $\mathcal{G}_{u,v}^L$ ) as histograms in Figure A.1. We observe both datasets follow long-tail distributions. Many subgraphs have tens of thousands of edges on DrugBank, while hundreds of thousands of edges on TWOSIDES since the DDI network is denser. Comparing with the augmented networks, whose sizes are 3,657,114 for DrugBank and 3,567,059 for TWOSIDES, the sizes of subgraphs are quite small.



**Fig. A.1** Histograms of subgraph sizes of  $\mathcal{G}_{u,v}^L$  (indicated by the number of edges) in the testing sets of two datasets when  $L = 3$ . Median value is indicated by the red dashed line.

## A.2 Evaluation metrics

As pointed by Yu et al (2021), there is at most one interaction between a pair of drugs in the DrugBank dataset (Wishart et al, 2018). Hence, we evaluate the performance in a multi-class setting, which estimates whether the model can correctly predict the interaction type for a pair of drugs. We consider the following metrics:

- F1(macro) =  $\frac{1}{\|\mathcal{P}_D\|} \sum_{p \in \mathcal{P}_D} \frac{2P_p \cdot R_p}{P_p + R_p}$ , where  $P_p$  and  $R_p$  are the precision and recall for the interaction type  $p$ , respectively. The macro F1 aggregates the fractions over different interaction types.

- Accuracy: the percentage of correctly predicted interaction type compared with the ground-truth interaction type.
- Cohen’s Kappa (Cohen, 1960):  $\kappa = \frac{A_p - A_e}{1 - A_e}$ , where  $A_p$  is the observed agreement (accuracy) and  $A_e$  is the probability of randomly seeing each class.

In the TWOSIDES dataset (Tatonetti et al, 2012), there may be multiple interactions between a pair of drugs, such as anaemia, nausea and pain. Hence, we model and evaluate the performance in a multi-label setting, where each type of side effect is modeled as a binary classification problem. Following (Zitnik et al, 2018; Tatonetti et al, 2012), we sample one negative drug pair for each  $(u, p, v) \in \mathcal{N}_{\text{test}}$  and evaluate the binary classification performance with the following metrics:

- ROC-AUC: the area under curve of receiver operating characteristics, measured by  $\sum_{k=1}^n \text{TP}_k \Delta \text{FP}_k$ , where  $(\text{TP}_k, \text{FP}_k)$  is the true-positive and false-positive of the  $k$ -th operating point.
- PR-AUC: the area under curve of precision-recall, measured by  $\sum_{k=1}^n P_k \Delta R_k$ , where  $(P_k, R_k)$  is the precision and recall of the  $k$ -th operating point.
- Accuracy: the average precision of drug pairs for each side effect.

### A.3 Hyperparameter selection

For all the baselines and the proposed EmerGNN, we use the hyper-parameter optimization toolbox hyperopt (Bergstra et al, 2015) to search for the optimum among 360 of hyper-parameter configurations. The objective of hyper-parameter selection is to maximize the premier metric performance (F1-score in DrugBank and PR-AUC in TWOSIDES) on the validation data. Adam (Kingma and Ba, 2014) is used as the optimizer to update the model parameters of EmerGNN. We list the tuned hyper-parameters and their ranges in Table A.4.

**Table A.4** Hyper-parameters and their tuning ranges for hyper-parameter selection.

Hyper-parameter	Ranges
Learning rate	$\{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}, 1 \times 10^{-2}\}$
Weight decay rate	$\{1 \times 10^{-8}, 1 \times 10^{-6}, 1 \times 10^{-4}, 1 \times 10^{-2}\}$
Mini-batch size	$\{32, 64, 128\}$
Representation size $d$	$\{32, 64\}$
Length of path-based subgraph $L$	$\{2, 3, 4\}$

### A.4 Implementation of baselines

In this part, we summarize the details of how baseline methods are implemented for the DDI prediction tasks.

- MLP (Rogers and Hahn, 2010). For each drug, there is a fingerprint vector with 1024 dimensions generated based on the drug’s SMILES attributes, which stand for Simplified Molecular Input Line Entry System. Given a pair of drugs  $u$  and  $v$ , the fingerprints  $\mathbf{f}_u$  and  $\mathbf{f}_v$  are firstly fed into an MLP with 3 layers, respectively. Then the representations are concatenated to compute the prediction logits in the same way as (4).
- Similarity (Vilar et al, 2014). We generate four fingerprints based on the SMILES representation for each drug. For a given pair of drugs, we compute the similarity features between this drug pair and a known set of DDIs. Specifically, we compare the 16 pairwise similarity features composed of the fingerprints of each drug pair, and select the maximum similarity value as the similarity feature for the current drug pair. Subsequently, we input these features into a random forest model to predict the DDIs.

- CSMDDI (Liu et al, 2022). CSMDDI uses a RESCAL-based method to obtain embedding representations of drugs and DDI types. It then utilizes partial least squares regression to learn a mapping function to bridge the drug attributes to their embeddings to predict DDIs. Finally, a random forest classifier is trained as the predictor, and the output of the random forest classifier provides the final prediction score for the interaction between two drugs. The implementation follows <https://github.com/itsosy/csmddi>.
- STNN-DDI (Yu et al, 2022). STNN-DDI learns a substructure×substructure×interaction tensor, which characterizes a substructure-substructure interaction (SSI) space, expanded by a series of rank-one tensors. According to a list of predefined substructures with PubChem fingerprint, two given drugs are embedded into this SSI space. A neural network is then constructed to discern the types of interactions triggered by the drugs and the likelihood of triggering a particular type of interaction. The implementation follows <https://github.com/zsy-9/STNN-DDI>.
- HIN-DDI (Tanvir et al, 2021). We constructs a heterogeneous information network (HIN) that integrates a biomedical network with DDIs. Within this network, we defined 48 distinct meta-paths, representing sequences of node types (including compounds, genes, and diseases) that connect nodes in the HIN. For each meta-path, a series of topological features, such as path count, was generated. Subsequently, these features were normalized and inputted into a random forest model for DDI prediction.
- MSTE (Yao et al, 2022). MSTE learns DDI with knowledge graph embedding technique and models the interactions as triplets in the KG. Specifically, for each interaction  $(u, i, v) \in \mathcal{N}_D$ , there are learnable embedding vectors  $\mathbf{e}_u, \mathbf{e}_v \in \mathbb{R}^d$  for the drugs  $u$  and  $v$ , respectively, and  $\mathbf{i} \in \mathbb{R}^d$  for interaction type  $i$ . MSTE then computes a score  $s(u, i, v) = \|\sin(\mathbf{i} \cdot \mathbf{e}_v) \cdot \mathbf{e}_u + \sin(\mathbf{e}_u \cdot \mathbf{e}_v) \cdot \mathbf{i} - \sin(\mathbf{e}_u \cdot \mathbf{i}) \cdot \mathbf{e}_v\|_{1/2}$ , which is then used as a negative logit for the prediction of interaction type  $i$ . The dimension  $d$  is a hyper-parameter tuned among {32, 64, 128}. The implementation follows <https://github.com/galaxysunwen/MSTE-master>.
- KG-DDI (Karim et al, 2019). KG-DDI uses a Conv-LSTM network on top of the embeddings to compute the score of interaction triplets  $(u, i, v) \in \mathcal{N}_D$  as well as the biomedical triplets  $(h, r, t) \in \mathcal{N}_B$ . Different from MSTE, KG-DDI firstly optimizes the parameters on both the interaction triplets and biomedical triplets, i.e., triplets in the augmented network, then fine-tunes on the interaction triplets for final prediction. The implementation follows <https://github.com/rezacsedu/Drug-Drug-Interaction-Prediction>.
- CompGCN (Vashishth et al, 2019). All the drugs, biomedical concepts, interactions and relations have their own learnable embeddings. These embeddings are aggregated by a graph neural network with 1 layer. The high-order embeddings  $\mathbf{h}_u^L, \mathbf{h}_v^L, \mathbf{h}_i^L$  are used to compute the score  $s(u, i, v) = \langle \mathbf{h}_u^L, \mathbf{h}_v^L, \mathbf{h}_i^L \rangle$ , which is then used as the logic of interaction type  $i$ . The implementation follows <https://github.com/mallabbiisc/CompGCN>.
- Decagon (Zitnik et al, 2018). Decagon is similar to CompGCN. The main difference is that the input biomedical network only considers biomedical concepts of drugs, genes and diseases, rather than the full biomedical network  $\mathcal{N}_B$ .
- KGNN (Lin et al, 2020). KGNN is built upon a GNN which propagates information and learns node representations within the new knowledge graph. Considering computational efficiency, KGNN employed neighbor sampling, with four neighbors sampled per layer for a total of two layers. Subsequently, the learned node representations were used to predict DDIs. The implementation follows <https://github.com/xzenglab/KGNN>.
- SumGNN (Yu et al, 2021). As discussed in Section 4.2, SumGNN has three steps. First, we extract enclosing subgraphs from the augmented network for all the drug pairs  $(u, v)$  to be predicted. Second, a node labeling trick is applied for all the enclosing subgraphs to compute the node features. Then, a graph neural network computes the graph representations of enclosing subgraphs, which are finally used to predict the interaction. The implementation follows <https://github.com/yueyu1030/SumGNN>.
- DeepLGF (Ren et al, 2022). The DeepLGF model contains three parts. First, the SMILES of drugs are used as sentences to encode the drugs' chemical structure. Second, a KG embedding model ComplEx is applied on the biomedical network to get the global embedding information of drugs. Third, a relational-GNN is used to aggregate the representations from the biomedical network. Finally, the three kinds of

representations are fused with an MLP module for the DDI prediction. Since there is no official code provided, we implement this model based on CompGCN.

## B Details of EmerGNN

### B.1 Algorithm for EmerGNN

In this part, we show the full algorithm and some implementation details of EmerGNN.

Given the augmented network  $\mathcal{N}$  and the drug pairs  $(u, v)$ , it will be time consuming to explicitly extract all the paths connecting  $u$  and  $v$  with length  $\leq L$ . In practice, we implicitly encode the pair-wise representations with Algorithm 1.

---

**Algorithm 1** EmerGNN: pair-wise representation learning with flow-based GNN.

---

```

1: procedure EMERGNN( $(u, v), L$ )            $\triangleright (u, v)$ : drug pair;  $L$ : the depth of path-based subgraph.
2:   initialize the  $u \rightarrow v$  pair-wise representation as  $\mathbf{h}_{u,e}^0 = \mathbf{f}_u$  if  $e = u$ , otherwise  $\mathbf{h}_{u,e}^0 = \mathbf{0}$ ;
3:   initialize the  $v \rightarrow u$  pair-wise representation as  $\mathbf{h}_{v,e}^0 = \mathbf{f}_v$  if  $e = v$ , otherwise  $\mathbf{h}_{v,e}^0 = \mathbf{0}$ ;
4:   for  $\ell \leftarrow 1$  to  $L$  do
5:     for  $e \in \mathcal{V}_D$  do                       $\triangleright$  This loop can work with matrix operations in parallel.
6:       message for  $u \rightarrow v$ :  $\mathbf{h}_{u,e}^{(\ell)} = \delta\left(\mathbf{W}^{(\ell)} \sum_{(e',r,e) \in \mathcal{N}_D} \sigma\left((\mathbf{w}_r^{(\ell)})^\top [\mathbf{f}_u; \mathbf{f}_v]\right) \cdot (\mathbf{h}_{u,e'}^{(\ell-1)} \odot \mathbf{h}_r^{(\ell)})\right)$ ;
7:       message for  $v \rightarrow u$ :  $\mathbf{h}_{v,e}^{(\ell)} = \delta\left(\mathbf{W}^{(\ell)} \sum_{(e',r,e) \in \mathcal{N}_D} \sigma\left((\mathbf{w}_r^{(\ell)})^\top [\mathbf{f}_u; \mathbf{f}_v]\right) \cdot (\mathbf{h}_{v,e'}^{(\ell-1)} \odot \mathbf{h}_r^{(\ell)})\right)$ ;
8:     end for
9:   end for
10:  Return  $\mathbf{W}_{\text{rel}}[\mathbf{h}_{u,v}^{(L)}; \mathbf{h}_{v,u}^{(L)}]$ .
11: end procedure

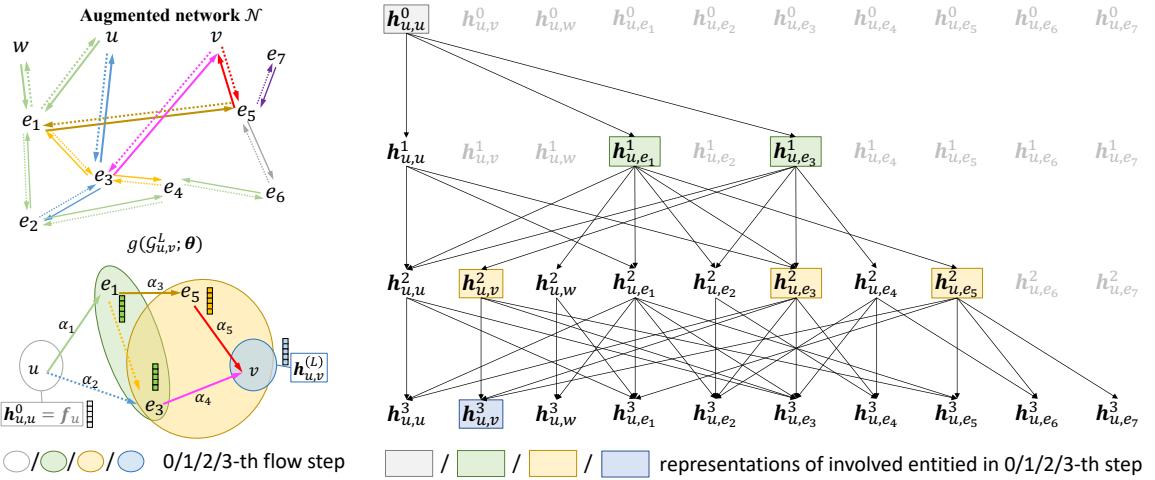
```

---

Take the direction  $u \rightarrow v$  as an example. Since we initialize  $\mathbf{h}_{u,e}^0 = \mathbf{f}_u$  if  $e = u$ , otherwise  $\mathbf{h}_{u,e}^0 = \mathbf{0}$  and the messages are computed based on a dot product operator  $\mathbf{h}_{u,e'}^{(\ell-1)} \odot \mathbf{h}_r^{(\ell)}$ , the representations of all entities with length longer than  $\ell$  away from  $u$  will be  $\mathbf{0}$  in the  $\ell$ -th step. In the end, only the entities with length  $\leq L$  will have valid representations. In addition, since we return  $\mathbf{h}_{u,v}^{(L)}$  for specific entity  $v$ , only the entities with length less than  $L - \ell$  can contribute to  $\mathbf{h}_{u,v}^{(L)}$  in the  $\ell$ -th step. In this way, we implicitly encode relevant entities and relations in the biomedical network from  $u$  to  $v$ .

To illustrate this point, we add the following example Figure B.1. The left part shows the symbolic representations of augmented network  $\mathcal{N}$  and the flow-based GNN in an explicit format. The right part shows how the representations are activated and propagated in each step.

- When  $\ell = 0$ , only  $\mathbf{h}_{u,u}^0$  is initialized with the non-zero features  $\mathbf{f}_u$  (in black) and other entities are initialized as  $\mathbf{0}$  (in gray).
- During the  $\ell$ -th iteration, the representations are flowed from  $u$  to the  $\ell$ -th hop neighbors of  $u$  in the  $\ell$ -th step (like the formulas in black, representing the representing a node in corresponding layer).
- At the last step  $\ell = L$ ,  $\mathbf{h}_{u,v}^L$  is used as the subgraph representation. We use boxes to indicate the representations participated in the computation of  $\mathbf{h}_{u,v}^L$  in each step.
- As shown, the entities in each step are identical to the entities in the left bottom figure, implicitly encoding the subgraph representation.



**Fig. B.1** A graphical illustration of why the initialization step together with the message propagation function can implicitly encode the visible entities in each layer (step  $\ell$ ). [Left]: symbolic representations of the example in Figure 1 of the main content. [Right]: representation flows according to the proposed Algorithm 1 (gray symbols mean  $\mathbf{0}$  vectors, and the relation types in lines are omitted for simplicity).

## B.2 Path extraction

Given a drug pair  $(u, v)$ , we use beam search to find the top  $B = 5$  paths connecting from  $u$  to  $v$  and top  $B = 5$  paths from  $v$  to  $u$ . Take the direction from  $u$  to  $v$  as an example. We provide the path extraction algorithm in Algorithm 2. We provide three kinds of lists: openList, recording the top  $K$  entities in each step; closeList, recording the accumulated scores of entities visited in each step; pathList, recording the searched paths at each step. In lines 3-4, we obtain the sets of entities visited in the  $\ell$ -th step  $\mathcal{V}_{u,v}^{(\ell)}$  through bi-directional bread-first-search. For each step, we compute the accumulated scores of entities  $e \in \mathcal{V}_{u,v}^{(\ell)}$  by summing the attention score  $\alpha_r^{(\ell)}$  in lines 7-8, and record the scores to the closeList. Then we pick up edges with top- $B$  scores, and add them to openList and pathList for next step computation in lines 11-13. After  $L$  steps, we aggregate the selected paths in pathList[1], ..., pathList[ $L$ ] to obtain the top- $B$  paths from  $u$  to  $v$ . The same steps are conducted to obtain the top- $B$  paths from  $v$  to  $u$ .

**Algorithm 2** Path extractor

---

```

1: procedure PATHEXTRACTOR( $(u, v), L, B$ )  $\triangleright B$ : the number of top paths in each direction.
2:   initialize openList[0]  $\leftarrow u$ ;
3:   set  $\mathcal{V}_{u,v}^{(0)} = \{u\}$ ,  $\mathcal{V}_{u,v}^{(L)} = \{v\}$ ;
4:   obtain  $\mathcal{V}_{u,v}^{(\ell)} = \{e : d(e, u) = \ell, d(e, v) = L - \ell\}, \ell = 1, \dots, L$  with bread-first-search;
5:   for  $\ell \leftarrow 1$  to  $L$  do
6:     set closeList[ $\ell$ ]  $\leftarrow \emptyset$ , pathList[ $\ell$ ]  $\leftarrow \emptyset$ ;
7:     for each edge in  $\{(e', r, e) : e' \in \text{openList}[\ell - 1], e \in \mathcal{V}_{u,v}^{\ell}\}$  do
8:       compute the attention weights  $\alpha_r^{(\ell)}$ ;
9:       compute score( $u, e', e$ )  $= \text{score}(u, e) + \alpha_r^{(\ell)}$ ;
10:      closeList[ $\ell$ ].add(( $e$ , score( $u, e', e$ )));
11:    end for
12:    for  $(u, e', e) \in \text{top}_B(\text{closeList}[\ell])$  do
13:      openList[ $\ell$ ].add( $e$ ), pathList[ $\ell$ ].add(( $e', r, e$ ));
14:    end for
15:  end for
16:  Return join(pathList[1]...pathList[ $L$ ]).
17: end procedure

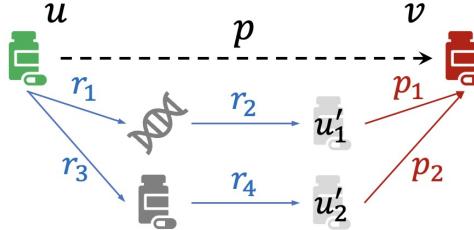
```

---

**B.3 Correlation computation**

The correlation matrix in Figure 3(a) is calculated as follows. Taking the interaction triplet  $(u, i_{\text{pred}}, v)$  in Figure B.2 as an example, we extract two paths which contain some relations in KG ( $r_1, \dots, r_4$ ) and interactions ( $i_1, i_2$ ) in  $\mathcal{N}$ . The co-occurrence times for each type  $i \in \mathcal{R}_I$  and  $r \in \mathcal{R}_B$  are counted on the paths for different interaction triplets.

For the interaction types  $i \in \mathcal{R}_I$  or biomedical relation types  $r \in \mathcal{R}_B$ , we group their counting values according to the to-be-predicted interaction  $i_{\text{pred}}$  and normalize the values by dividing the frequency of  $i_{\text{pred}}$  in  $\mathcal{N}_{\text{D-test}}$ .



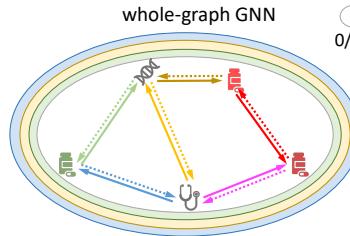
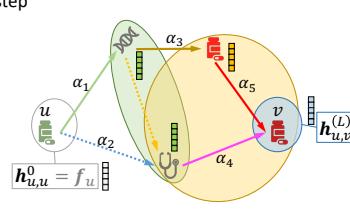
**Fig. B.2** A graphical exemplar of selected paths.

## B.4 Detailed comparison between SumGNN and EmerGNN

In this part, we further analyze the difference between SumGNN and EmerGNN in terms of the following four aspects:

- Subgraph: The enclosing subgraph used in SumGNN contains all the edges among entities within  $L$  steps away from both  $u$  and  $v$ ; the path-based subgraph only considers edges pointing from  $u$  to  $v$  or  $v$  to  $u$ .
- Node labeling: SumGNN requires a node labeling procedure to compute the distance of nodes on the subgraph to the target drugs  $u$  and  $v$ ; however, as the edges are connected in the direction from  $u$  to  $v$ , in EmerGNN, the distance information can be reflected by the number of jumps, thus EmerGNN does not need a node labeling procedure.
- GNN architecture: SumGNN uses the whole-graph GNN as in (Zhang and Chen, 2018; Xu et al., 2018; Teru et al., 2020) to propagate over the whole subgraph. EmerGNN uses the flow-based GNN to propagate information from  $u$  to  $v$  step-by-step with a better control of information flow.
- Pooling: In SumGNN, the representations of all the entities in the subgraph should be pooled (e.g., concatenated) for final interaction prediction; however, benefiting from the flowing pattern of flow-based GNN, all the information can be ordered and integrated when propagating from  $u$  to  $v$ , thus EmerGNN only uses the final step representation of  $v$ , i.e.,  $\mathbf{h}_{u,v}^{(L)}$ , for interaction prediction.

Since EmerGNN takes better care of the information on biomedical network and the order of information flow from an emerging drug to an existing drug than SumGNN, it empirically performs better in both settings in Table 1.

Subgraph	enclosing subgraph	path-based subgraph
Node labeling	needed	not needed
GNN architecture	whole-graph GNN 	flow-based GNN 
Pooling	Representations of all the entities in subgraph $\{\mathbf{h}_e^{(L)}, e \in \mathcal{V}_{u,v}^L\}$ are pooled for prediction	$\mathbf{h}_{u,v}^{(L)}$ is directly used for prediction

**Fig. B.3** A detailed comparison between SumGNN and EmerGNN in terms of subgraph and GNN architecture design.

## C Additional Results

### C.1 Performance comparison of the S0 setting

There are three basic settings for the DDI prediction ([Dewulf et al, 2021](#); [Liu et al, 2022](#); [Yu et al, 2022](#)): (S0) interaction between existing drugs; (S1) interaction between emerging and existing drugs; and (S2) interaction between emerging drugs.

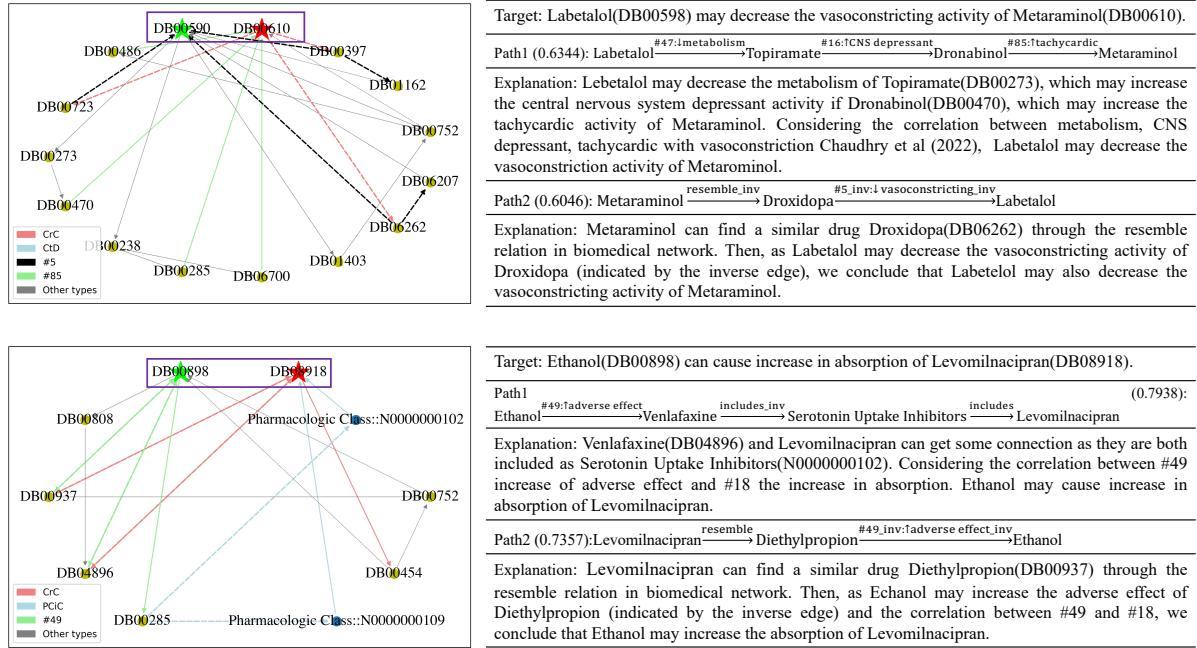
**EmerGNN** has shown significant advantage over the baseline methods for emerging drug prediction in Table 1. We also compare the performance in the S0 setting for prediction between existing drugs in Table C.1, where the setting exactly follows [Yu et al \(2021\)](#). Comparing the two tables, we find that the emerging drug prediction task is much harder than existing drug prediction as the accuracy values in the Table 1 are much lower than those in Table C.1. Even though the shallow models MLP, **Similarity**, HIN-DDI perform well in predicting DDIs for emerging drugs, they are worse than the deep networks when predicting DDIs between existing drugs. The embedding model **MSTE** performs very poorly for emerging drugs but is the third best for existing drug prediction. The GNN-based methods, especially **SumGNN**, also works well for predicting DDIs between existing drugs. This demonstrates that drug embeddings and deep networks can be helpful for drug interaction prediction if sufficient data are provided. **EmerGNN**, even though specially designed for emerging drug prediction, still outperforms the baselines with a large margin for predicting interactions between existing drugs. These results again show the flexibility and strengths of **EmerGNN** on the DDI prediction task.

**Table C.1** Comparison of different methods on the DDI prediction between two existing drugs. “FF” is short for “Fingerprint Feature”; “GF” is short for “Graph Feature”; “Emb” is short for “Embedding”; and “GNN” is short for “Graph Neural Network”. The evaluation metrics are presented in percentage (%) with best values in boldface and the second best underlined.

Datasets (S0 setting)		DrugBank			TWOSIDES		
Type	Methods	F1-Score	Accuracy	Kappa	PR-AUC	ROC-AUC	Accuracy
DF	MLP ( <a href="#">Rogers and Hahn, 2010</a> )	61.1±0.4	82.1±0.3	80.5±0.2	81.2±0.1	82.6±0.3	73.5±0.3
	Similarity ( <a href="#">Vilar et al, 2014</a> )	55.0±0.3	62.8±0.1	67.6±0.1	59.5±0.0	59.8±0.0	57.0±0.1
GF	HIN-DDI ( <a href="#">Tanvir et al, 2021</a> )	46.1±0.5	54.4±0.1	63.4±0.1	83.5±0.2	87.7±0.3	82.4±0.3
Emb	MSTE ( <a href="#">Yao et al, 2022</a> )	83.0±1.3	85.4±0.7	82.8±0.8	90.2±0.1	91.3±0.1	84.1±0.1
	KG-DDI ( <a href="#">Karim et al, 2019</a> )	52.2±1.1	61.5±2.8	55.9±2.8	88.2±0.1	90.7±0.1	83.5±0.1
GNN	CompGCN ( <a href="#">Vashisht et al, 2019</a> )	74.3±1.2	78.8±0.9	75.0±1.1	90.6±0.3	92.3±0.3	84.8±0.3
	Decagon ( <a href="#">Zitnik et al, 2018</a> )	57.4±0.3	87.2±0.3	86.1±0.1	90.6±0.1	91.7±0.1	82.1±0.5
	KGNN ( <a href="#">Lin et al, 2020</a> )	74.0±0.1	90.9±0.2	89.6±0.2	90.8±0.2	92.8±0.1	86.1±0.1
	SumGNN ( <a href="#">Yu et al, 2021</a> )	86.9±0.4	92.7±0.1	90.7±0.1	93.4±0.1	94.9±0.2	88.8±0.2
EmerGNN		<b>94.4±0.7</b>	<b>97.5±0.1</b>	<b>96.6±0.8</b>	<b>97.6±0.1</b>	<b>98.1±0.1</b>	<b>93.8±0.2</b>
p-values		4.5E-7	6.5E-13	6.7E-8	2.3E-8	5.1E-10	6.1E-7

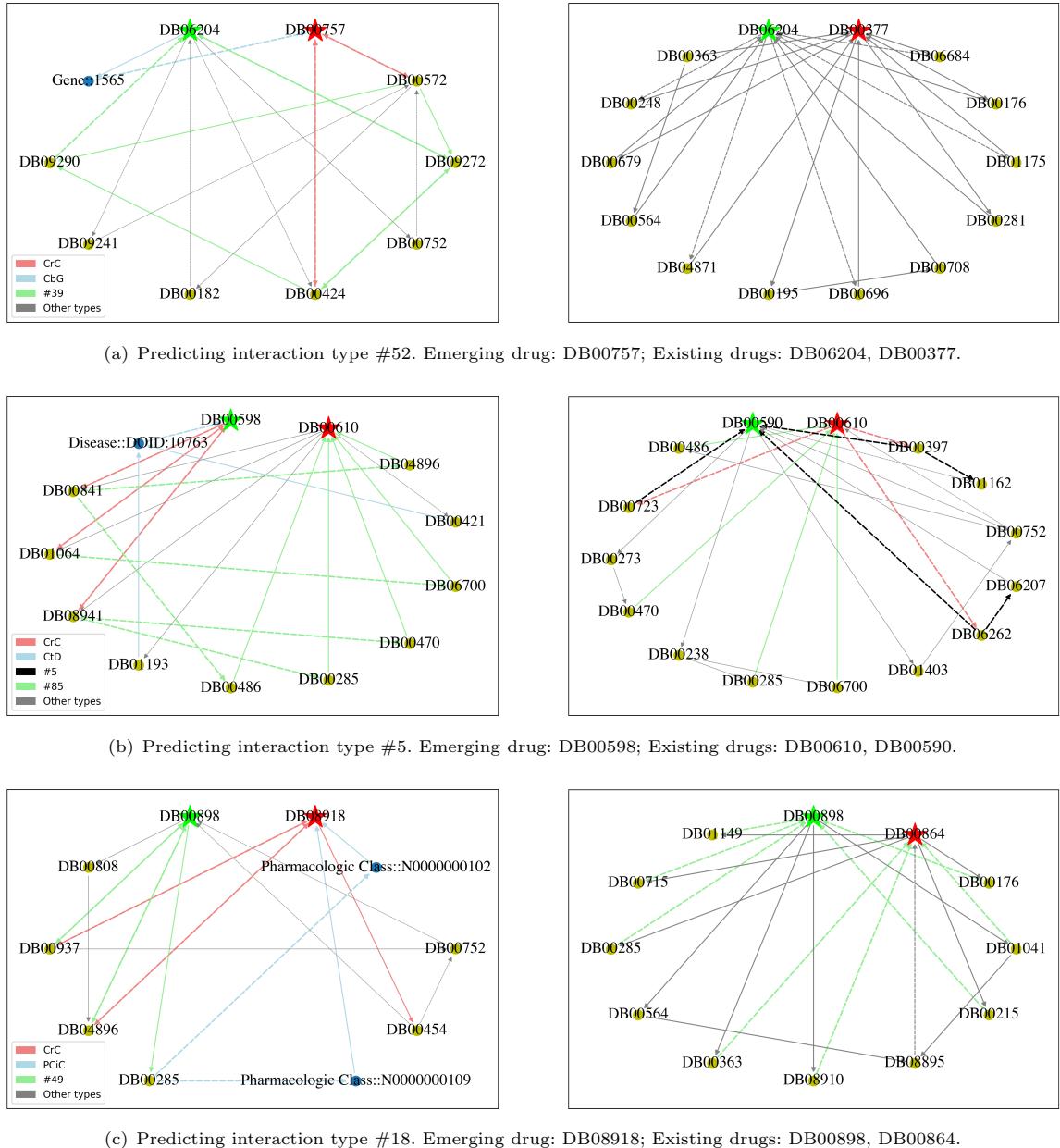
## C.2 Path visualization

In Figure C.1, we provide the more examples with detailed path explanation from the learned model.



**Fig. C.1** Additional results for detailed path explanation.

In Figure C.2, we provide additional results for path visualization between the case of S1 setting and S0 setting. Specifically, we choose examples with predicted interaction types #52, #5 and #18 in Figure 3(c). We plot the interactions between emerging and existing drugs in the left part, and the interactions between two existing drugs in the right part. As shown in Figure C.2, relation type CrC plays an important role during prediction, which is also reflected by the high correlations in Figure 3(d). For the interaction types on the subgraphs, we also observe the correlations of interaction types, i.e., (#52, #39), (#5, #85) and (#18, #49), which are identified in Figure 3(c). Comparing the left part with right part, we observe that the biomedical entities, like Gene::1565, Disease::DOID:10763 and Pharmacologic Class::N000000102(9), play the role to connect the emerging drug and existing drug. However, the prediction of interactions between two existing drugs relies mainly on the DDI between drugs. These results again verify the claim that EmerGNN is able to identify and leverage the relevant entities and relations in the biomedical network.



**Fig. C.2** Visualization of learned paths between drug pairs. Left parts: interactions between an existing drug and an emerging drug. Right parts: interactions between two existing drugs. The dashed lines mean inverse types.