KGNN: Knowledge Graph Neural Network for Drug-Drug Interaction Prediction

Xuan Lin[1] Zhe Quan[1,*] Zhi-Jie Wang[2,*] Tengfei Ma[1] Xiangxiang Zeng[1]

[1]Hunan University [2]Chongqing University

(* indicates corresponding author)

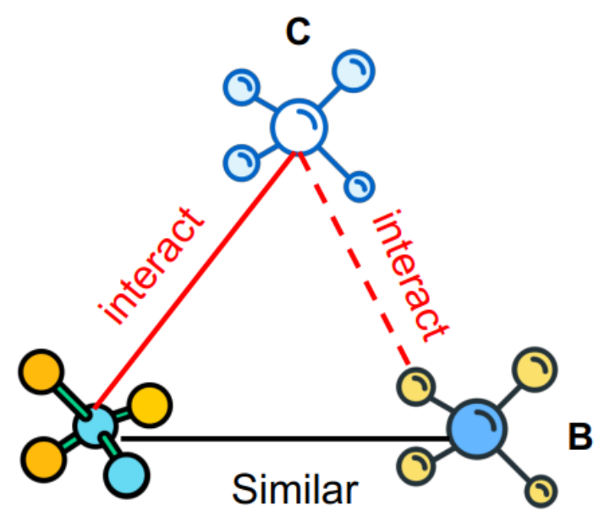
IJCAI-PRICAI
YOKOHAMA
2020

Problem Definition and Contribution

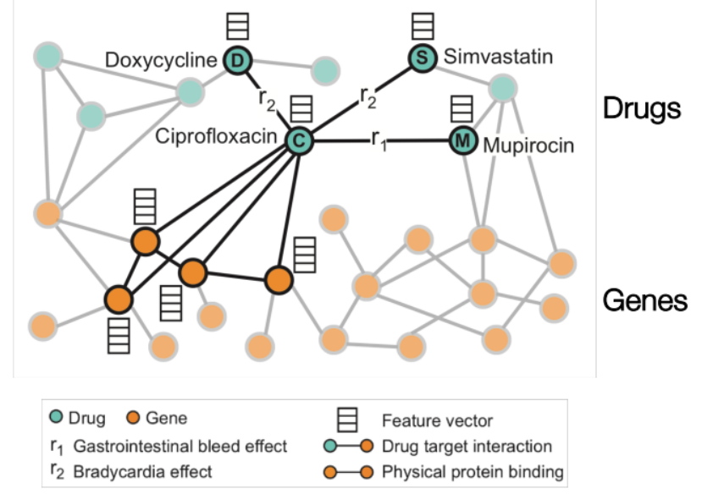
Goal: Effectively identifying potential drug-drug interactions (DDIs) during clinical trials is critical for patients and society.

Motivations:

- **Similarity-based**: drugs with similar representations will perform similar DDIs, this line of work needs design specialized drug representation.



Similarity



Network embedding

- **Network embedding-based**: drug combination leads to polypharmacy side effect, while this line of work only focus on single relation.

Key Contributions:

- An effective and novel framework for drug-drug interaction prediction.
- Extends spatial-based GNN methods to the knowledge graph.
- Provides new insights into the study of jointly considering topological structure information of drug and semantic relation of knowledge graph.

Problem Formulation

Problem Definition: to automatically capture both high-order structures and semantic relations in knowledge graph, we formulate the knowledge graph-based DDI prediction problem from two aspects as follows.

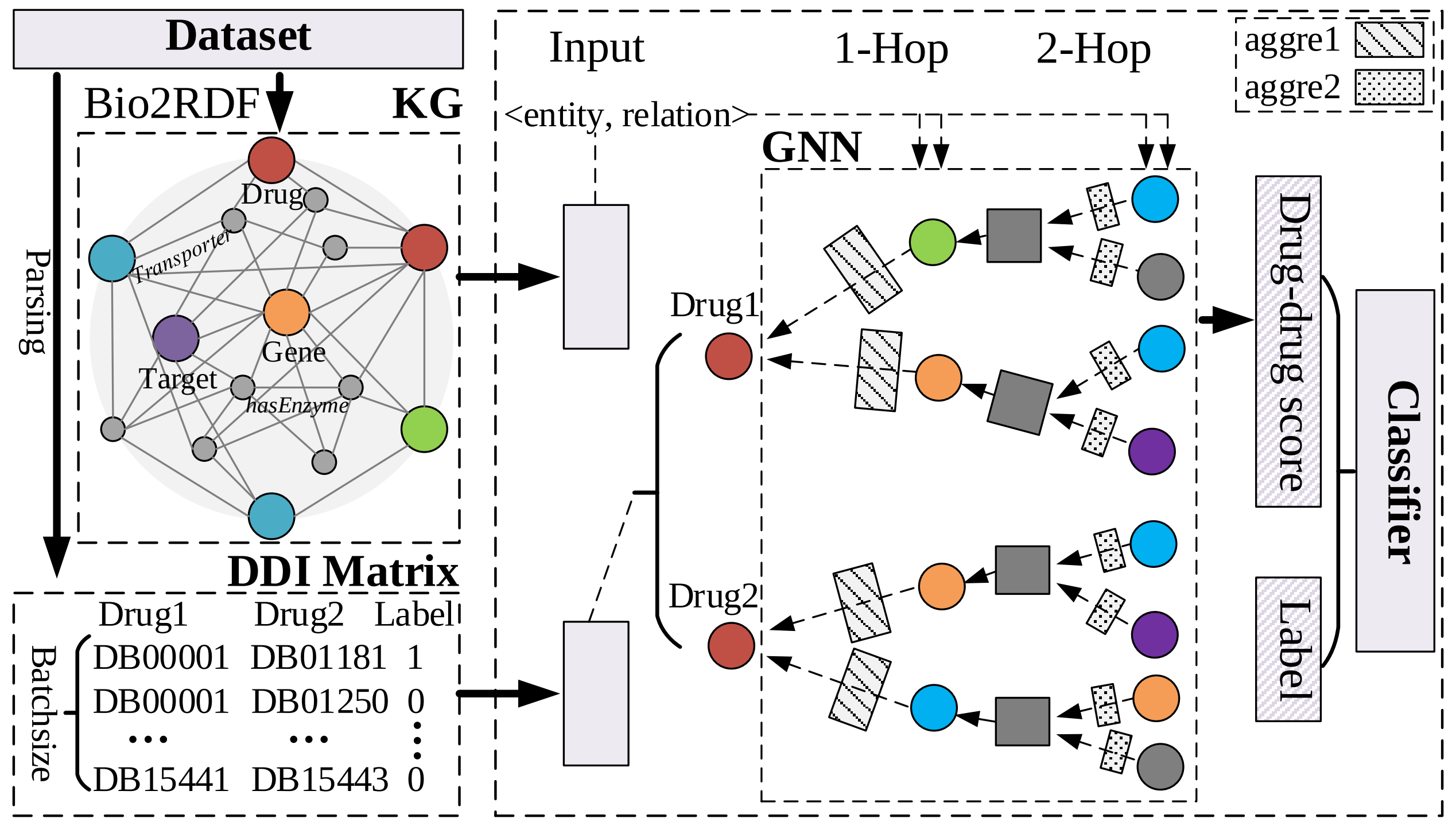
DDI matrix: given a set N_d of drugs, and define the drug-drug interaction matrix $Y \in (0,1)^{|N_d| \times |N_d|}$, where $|N_d|$ denotes the number of drugs. In the matrix, for each entry $y_{i,j} = 1$ $(i, j \in N_d, j \neq i)$, if its value is 1, then it means that drug j interacts with drug i.

Knowledge graph: considering neighborhood topologies for drug related entities (e.g., targets), in the form of knowledge graph. We denote a KG by $G = (N_e, N_r)$, which is comprised of entity-relation-entity triples, where N_e (resp., N_r) is the set of entities (resp., relations). For any knowledge graph triple $T_i = (h_i, r_i, t_i)$, where $h_i, t_i \in N_e$ and $r_i \in N_r$, it describes a relationship of type r_i between entity h_i and t_i.

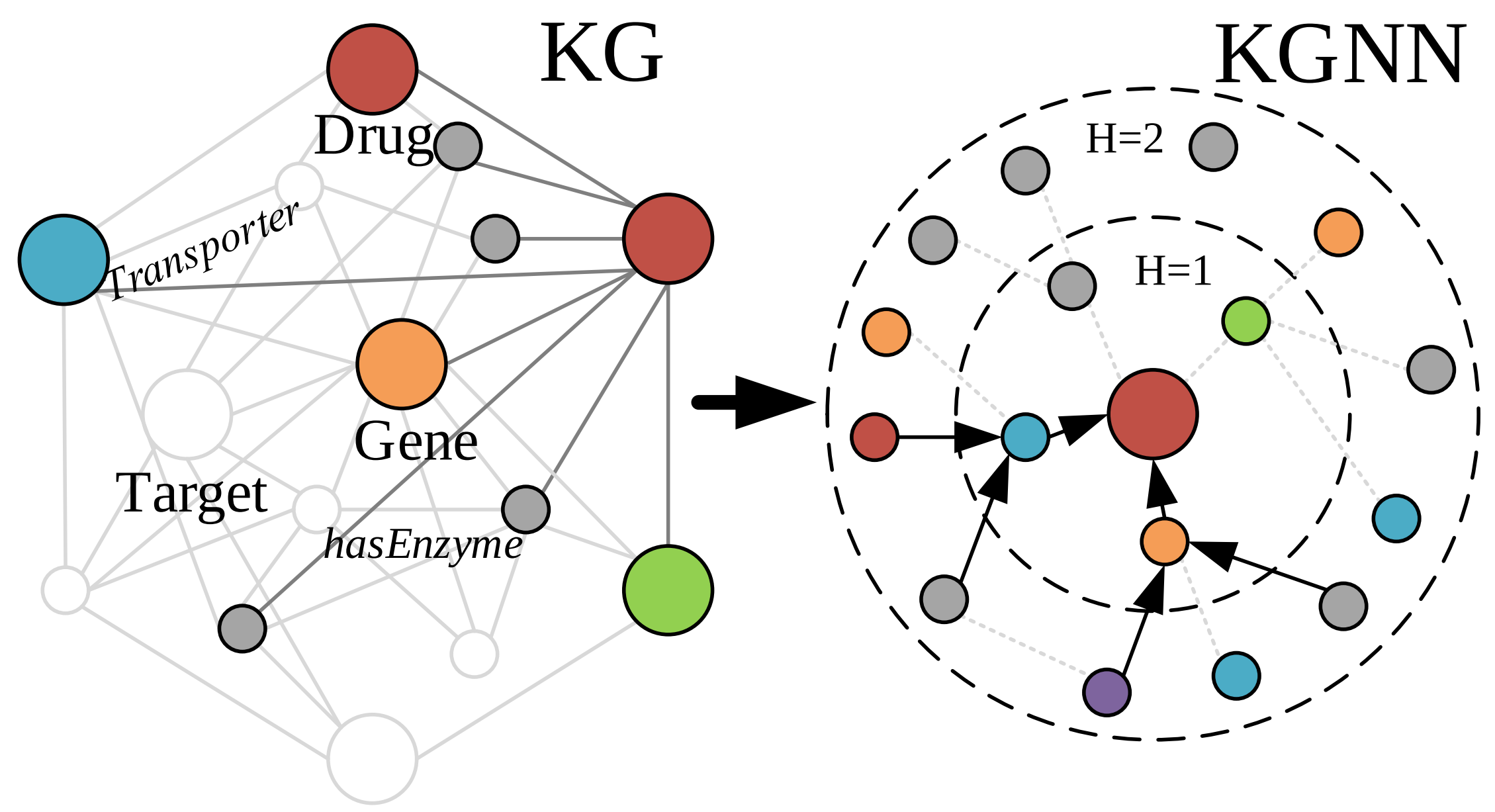
Main Idea: given the DDI matrix Y and the knowledge graph G, we aim to predict whether drug $i (i \in N_d)$ has potential interaction with drug $j (j \in N_d, j \neq i)$, while such an interaction has not been discovered before. To achieve it, our main task is to learn a prediction function $\hat{y}_{i,j} = \Gamma(i, j | \beta, Y, G)$, where $\hat{y}_{i,j}$ denotes the probability that drug j will interact with drug i, and β denotes the model parameters of function Γ.

Method

KGNN comprises three parts, namely DDI Extraction and KG Construction, KGNN Layer and DDI classifier.



The overall of proposed KGNN



KGNN layer

Loss Function:

$$\mathcal{L}oss = \sum_{(i,j) \in Y (i,j \in N_d, j \neq i)} -y_{i,j} log y_{i,j} - (1 - y_{i,j}) log(1 - \hat{y}_{i,j}) \tag{1}$$

where $\hat{y}_{i,j}$ is the predicted value, $y_{i,j}$ is the ground-truth value, and Y represents the set of drug-drug pairs.

Experiments & Results

Dataset:

We evaluate our proposed KGNN[a] by using two datasets.

- **DrugBank**: we parse the verified DDIs of the provided profile from DrugBank (V5.1.4) and compile an edge list of drug identifier combinations, which obtains 2,578 approved small molecule drugs and 612,388 unique approved DDIs spanning 13,339 drugs;
- **KEGG-drug**: we parse the sources from KEGG and map it to DrugBank identifiers (IDs), which results in 1,925 approved drugs and 56,983 approved interactions spanning 11,147 drugs and 324,183 interactions.

The statistic of two widely used datasets

	DrugBank	KEGG-drug
Drugs	2,578	1,925
Interactions	612,388	56,983
Entities	2,129,712	129,910
Relation Types	72	167
KG Triples	7,852,852	362,870

[a]https://github.com/xzenglab/KGNN

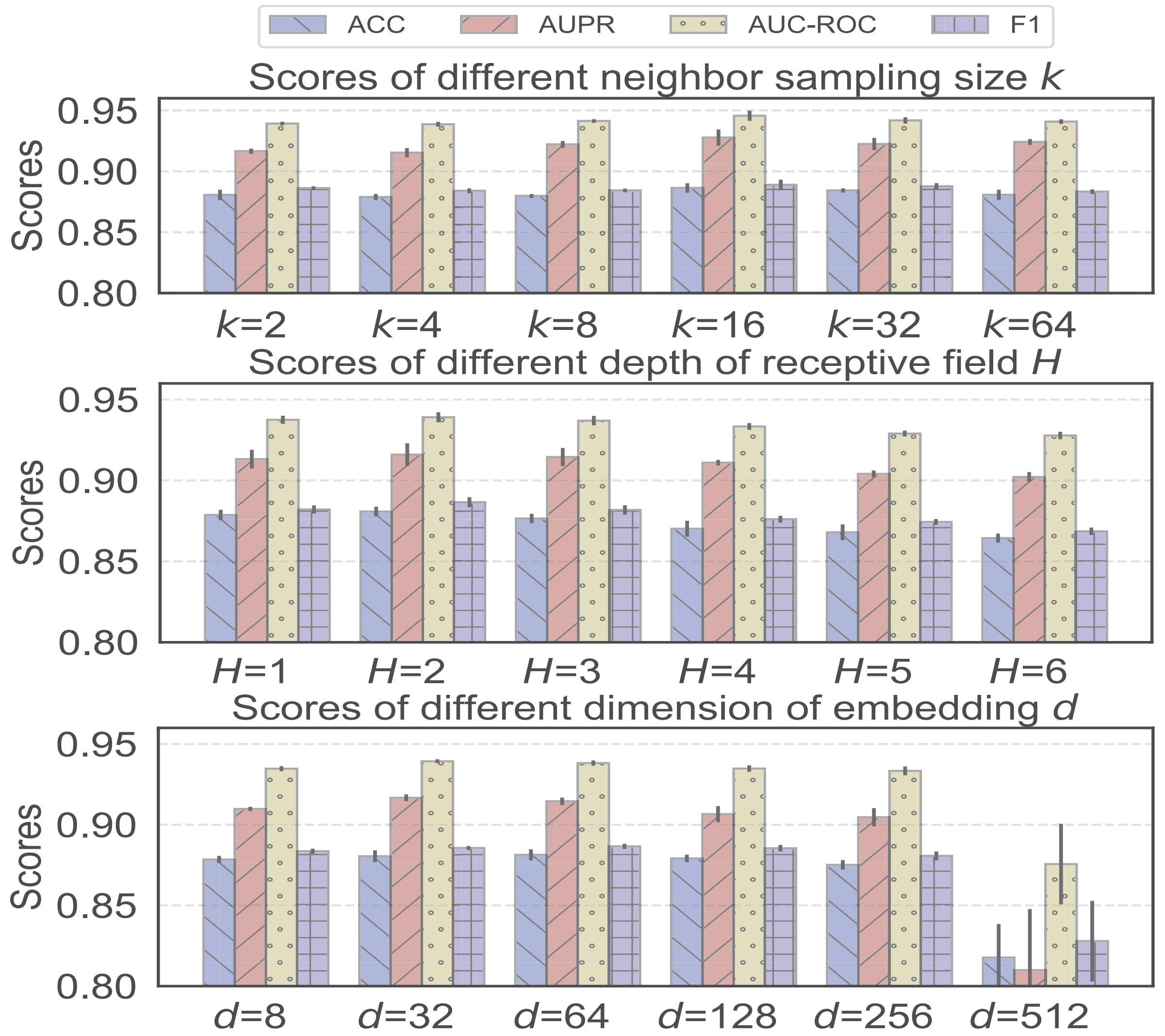
Baselines:

- **Matrix Factorization:** Laplacian [1] and GraRep [2] aim to factorize the matrix of input data into lower dimensional matrices.
- **Random Walk:** DeepWalk [3] and Struc2Vec [4] learn node representations by generating node sequences and by using different random walks strategies in graphs.
- **Neural Network:** LINE [5], SDNE [6] and GAE [7] adopt different neural architectures and different kinds of graph information as input to learn node embedding.
- **DeepDDI:** [8] develops a deep learning-based method that reduces the dimension of drug features, based on a principal component analysis.
- **KG-ddi:** [9] is a KG-based method for DDI prediction, by encompassing over 12,000 drug features from integrated knowledge graph of multiple data sources, and it adopts a CNN-LSTM model using the embeddings.
- **KGNN$_x$:** represents the variants of our proposed KGNN, where the subscript x attached denotes different aggregation operations.

Performance of KGNN Against Comparative Approaches:

Methods / Metrics	MF-based		RW-based		NN-based			DL-based	KG-based	KGNN$_x$		
	Laplacian	GreRep	DeepWalk	struc2vec	LINE	SDNE	GAE	DeepDDI	KG-ddi	*neighbor*	*sum*	*concat*
ACC	0.7183 0.8029	0.8443 0.8718	0.8349 0.8547	0.7882 0.8436	0.8280 0.8655	0.8303 0.8674	0.7491 0.7586	0.8123 0.8229	0.7867 0.8154	0.9354 0.8846	0.9538 0.8882	**0.9561** **0.8950**
AUPR	0.7533 0.8261	0.9115 0.9055	0.9070 0.9011	0.8672 0.8861	0.8915 0.8968	0.8782 0.8967	0.7403 0.7571	0.9193 0.8442	– –	0.9801 0.9207	0.9890 0.9247	**0.9892** **0.9533**
AUC-ROC	0.7966 0.8736	0.9230 0.9305	0.9181 0.9208	0.8735 0.9086	0.9092 0.9264	0.9029 0.9249	0.8085 0.8334	0.9261 0.8994	0.7867 0.8154	0.9824 0.9418	0.9902 0.9453	**0.9912** **0.9518**
F1	0.7270 0.8079	0.8461 0.8748	0.8357 0.8570	0.7962 0.8476	0.8318 0.8695	0.8373 0.8704	0.7889 0.7888	0.8466 0.7966	0.7843 0.8152	0.9366 0.8869	0.9544 0.8909	**0.9566** **0.8982**

Results of KGNN with Varying Size of k, H, and d:



Impact of neighborhood size: vary the neighborhood size k and KGNN achieves the best performance when k = 16.

Impact of depth of receptive field: investigate the influence of depth of receptive field H by varying from 1 to 6, and the performance of our model in all metrics decreases starting from H = 3.

Impact of dimension of embedding: it can boost the performance with a proper d (d=32 or 64).

References:

[1] Laplacian [Belkin and Niyogi, Neural computation 2003]
[2] GraRep [Cao *et al.*, CIKM 2015]
[3] DeepWalk [Perozzi *et al.*, KDD 2014]
[4] Struc2Vec [Ribeiro *et al.*, KDD 2017]
[5] LINE [Tang *et al.*, WWW 2015]
[6] SDNE [Wang *et al.*, KDD 2016]
[7] GAE [Kipf and Welling, NeurIPS 2016]
[8] DeepDDI [Ryu *et al.*, PNAS 2018]
[9] KG-ddi [Karim *et al.*, ACM BCB 2019]