Central-Smoothing Hypergraph Neural Networks for Predicting Drug–Drug Interactions

Duc Anh Nguyen, Canh Hao Nguyen, and Hiroshi Mamitsuka, *Senior Member, IEEE*

***Abstract*— Predicting drug–drug interactions (DDIs) is the problem of predicting side effects (unwanted outcomes) of a pair of drugs using drug information and known side effects of many pairs. This problem can be formulated as predicting labels (i.e., side effects) for each pair of nodes in a DDI graph, of which nodes are drugs and edges are interacting drugs with known labels. State-of-the-art methods for this problem are graph neural networks (GNNs), which leverage neighborhood information in the graph to learn node representations. For DDI, however, there are many labels with complicated relationships due to the nature of side effects. Usual GNNs often fix labels as one-hot vectors that do not reflect label relationships and potentially do not obtain the highest performance in the difficult cases of infrequent labels. In this brief, we formulate DDI as a hypergraph where each hyperedge is a triple: two nodes for drugs and one node for a label. We then present CentSmoothie, a hypergraph neural network (HGNN) that *learns representations of nodes and labels altogether* with a novel "*central-smoothing*" formulation. We empirically demonstrate the performance advantages of CentSmoothie in simulations as well as real datasets.**

***Index Terms*— Drug–drug interactions (DDIs), hypergraph Laplacian, hypergraph neural networks (HGNNs), smoothing.**

I. Introduction

In drug–drug interactions (DDIs), concurrent use of two drugs can lead to side effects, which are unwanted reactions in human bodies. It is a very important task to predict DDIs to guide drug safety. Given drug information and known side effects of many pairs of drugs, one wishes to learn a model to predict side effects of all pairs of drugs, which include new pairs of drugs without known side effects or known pairs (to denoise or complete side-effect data). DDI is usually represented as a graph with nodes for drugs, edges for drug pairs that interact, with (binary vector) labels for (known) side effects [1]. The task is to predict labels of all pairs of nodes in the DDI graph. Fig. 1(a) shows an example of a DDI graph, where the dotted edge with question marks is the pair of drugs with labels to be predicted.

Recently, graph neural networks (GNNs) have emerged as a prominent approach for this task with high prediction performance [1], [2]. GNNs for predicting DDI have two steps: learning new representations of drugs from a DDI graph and using these representations for predictions. One drawback of this approach is the lack of learning label (i.e., side effect) representations. There are many side effects with complicated relationships. For example, our largest dataset has 964 side effects, where the number of drug pairs for one side effect (positive samples in supervised learning) ranges from 288 to 22 520. Previous methods represent each side effect as an independent one-hot vector, potentially underutilizing the relationship among side effects [1], [2], [3]. Considering the relationship between side effects would be beneficial for predicting side effects, especially the ones with only small numbers of positive samples (i.e., infrequent side effects). Hence, it is desirable to learn the representations for both drugs and side effects, namely, both nodes and edge labels, together.

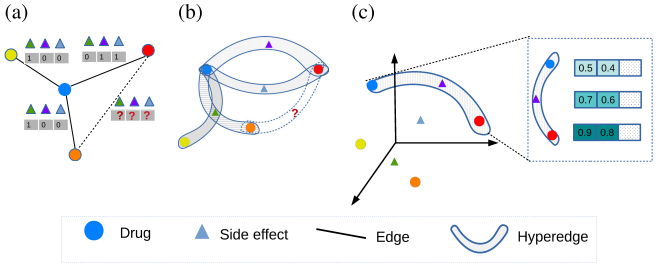

Fig. 1. Illustrative examples of (a) traditional graph, (b) (proposed) hypergraph for DDIs, and (c) central-smoothing assumption.

To this end, we propose to encode DDI data with a hypergraph [4]. A node in the hypergraph can be either a drug or a side effect. A hyperedge is a triple of two drugs and a side effect that they caused. Hence, a pair of drugs with multiple side effects (interactions) will result in many hyperedges in the hypergraph. Fig. 1(b) illustrates an example of a hypergraph corresponding to the DDI graph in Fig. 1(a). The existing learning methods of hypergraph neural networks (HGNNs) are based on a *smoothing assumption* that the representations of nodes in a hyperedge should be close to each other [5], [6]. However, this assumption is not necessarily appropriate for our DDI problem, since each node representation should reflect the (chemical or biological) properties of the corresponding drug, and interacting drugs do not necessarily need to have similar properties.

We propose CentSmoothie, a central-smoothing HGNN that uses our idea, *central-smoothing assumption* [see Fig. 1(c)] for each hyperedge in the hypergraph for DDI. The idea is to learn k-dimensional representation vectors for nodes in a hyperedge such that the following hold: 1) a drug node representation reflects the property of the corresponding drug and 2) a side-effect node representation reflects a combination of some properties of the two drugs that cause the corresponding side effect [7]. To implement 2), we first assume that a side-effect representation should be related to the midpoint of the representations of the two interacting drugs, reflecting the combination of the two drug properties. Furthermore, there might have different side effects of the same two drugs, suggesting that each side effect might be obtained by a partial combination of the two drug properties. Hence, we propose that the representation for each side effect is learned to be close to a weighted midpoint of the corresponding two drug representations.

Manuscript received 17 September 2021; revised 24 April 2022 and 4 August 2022; accepted 21 March 2023. The work of Duc Anh Nguyen was supported in part by the Otsuka Toshimi Scholarship Foundation. The work of Canh Hao Nguyen was supported in part by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) KAKENHI under Grant 22K12150. The work of Hiroshi Mamitsuka was supported in part by MEXT KAKENHI under Grant 16H02868, Grant 19H04169, Grant 21H05027, and Grant 22H03645; and in part by the Artificial Intelligence in Physical Sciences and Engineering Research (AIPSE) Program through the Academy of Finland. *(Corresponding author: Duc Anh Nguyen.)*

Duc Anh Nguyen and Canh Hao Nguyen are with the Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan (e-mail: ducanh@kuicr.kyoto-u.ac.jp).

Hiroshi Mamitsuka is with the Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan, and also with the Department of Computer Science, Aalto University, 02150 Espoo, Finland.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3261860.

Digital Object Identifier 10.1109/TNNLS.2023.3261860

We formulate the above assumption, and then define the central-smoothing hypergraph Laplacian to be used in each layer of the HGNN with spectral convolution [5]. We also provide a computational method with the complexity of $O(n)$ for the proposed hypergraph Laplacian.

We conducted extensive experiments to verify the performance advantages of CentSmoothie in both synthetic and real datasets. Our experimental results demonstrated that CentSmoothie significantly outperformed the existing spectral-based convolutional HGNNs in all cases. In particular, CentSmoothie achieved higher performances over baselines for real datasets with more infrequent side effects, which are more difficult to predict, justifying the benefit of learning label (side effect) representations.

II. Related Work

The existing work in predicting DDI can be divided into two approaches: non-graph-based and graph-based ones. In the non-graph-based approach, predefined feature vectors, indicating the existences of chemical substructures and interacting proteins of drugs, are used. The side effects can be predicted by using a model [for example, a multilayer feedforward neural network (MLNN)], which receives the feature vectors of two drugs as input and the vector indicating the side effects of the two drugs as output [3], [8].

In the graph-based approach, topological information of graphs is used to enhance the representations of nodes, leading to higher performance than the non-graph-based approach. There are two types of graphs that can be used: molecular graphs of drugs and a DDI graph. For a DDI graph where nodes are drugs and edges are interactions between drugs, GNNs are applied to learn a new representation of a drug node based on its neighbors. Recent results show that GNNs for predicting DDI achieve the cutting-edge performance [1], [2]. An extension of a DDI graph can be a DDI heterogeneous graph, where nodes are drugs, and side effects and edges are pairs of interacting drugs or drug-side effects [9]. However, the DDI heterogeneous graph cannot preserve triples of drug–drug-side effects.

GNNs can be further divided into two approaches: spectral convolution and spatial convolution [10]. In the spectral convolution, at first, the graph Laplacian is defined, and then, each GNN layer is constructed from the graph Fourier transformation given the graph Laplacian [5], [11]. The spatial convolution approach uses node spatial relation that a node is updated based on information from neighbor nodes [9], [12].

Different from the existing work for predicting DDIs, we formulate the DDIs in the form of a hypergraph and develop a new HGNN on the DDI hypergraph.

In HGNNs, recent work has inherited the spectral convolution approach on graphs to adapt to hypergraphs by defining the hypergraph Laplacian [5]. Once the hypergraph Laplacian is defined, HGNNs can be constructed in the same manner as that for GNNs. Another approach for HGNNs is the spatial convolution approach with attention mechanisms [6].

III. Background

In this section we briefly describe the hypergraph Laplacian being derived from a smoothness measure [4]. Let $G = (V, E)$ be a general hypergraph, where V is the node set and $E \subset 2^V$ is the hyperedge set. Let $W = \text{diag}(w(e_1), \ldots, w(e_{|E|})) \in \mathbb{R}^{|E|\times|E|} \succcurlyeq \mathbf{0}$ be the diagonal matrix that $w(e)$ is the weight of hyperedge e. Let $x \in R^{|V|}$ be the values of nodes on the hypergraph that x_u is the value of x at node u.

The hypergraph Laplacian is usually defined to be used in a similar manner to the graph Laplacian: to evaluate the smoothness of a function on a graph. Let $\text{sh}(x, G)$ be a smoothness measure of x on G, and $\text{ss}(x, e)$ be a smoothness measure of x on hyperedge e. The smoothness on the hypergraph usually has the following form [4]:

$$\text{sh}(x, G) = \mathcal{T}_{e\in E} w(e)\text{ss}(x, e) \tag{1}$$

where $\mathcal{T}$ is an aggregation operator, such as sum (the most commonly used one), max, or l_p norm [4]. Usual smoothing assumption on hypergraphs is that nodes within a hyperedge should be close to each other [5], [6], [13], and then, the smoothness measure on each hyperedge is calculated by

$$\text{ss}(x, e) = \sum_{(u,v)\in e} (x_u - x_v)^2. \tag{2}$$

When $\mathcal{T}$ is a sum operator, the smoothness of a function on a hypergraph can be found in the following form:

$$\text{sh}(x, G) = \sum_{e\in E} w(e) \sum_{(u,v)\in e} (x_u - x_v)^2 = x^{\mathbf{T}} L x \tag{3}$$

which has the quadratic form with L, and L is then called the hypergraph Laplacian of the hypergraph. In the next section, we will propose a new smoothing assumption on hypergraphs and then define a new hypergraph Laplacian.

IV. CentSmoothie (Central-Smoothing HGNNs)

A. Problem Setting

We formulate the problem of predicting DDI as follows.

Input: Given a hypergraph of DDIs: $G = (V, E)$, where the node set $V = V_D \cup V_S$ consists of a drug node set V_D and a side-effect node set V_S, a known hyperedge set $E \subset V_D \times V_D \times V_S$ (since two drugs in a drug pair are unordered, two triples (u, v, t) and (v, u, t) $(u, v \in V_D$ and $t \in V_S)$ are the same), and the feature vectors of drugs: $X_D \in R^{|V_D|\times K_0}$, where K_0 is the feature size. The feature vectors of side effects are one-hot vectors.

Output: For each triple $e = (u, v, t) \in V_D \times V_D \times V_S$, t is predicted to be a side effect of u and v if the score of the triple is larger than a threshold.

B. Central-Smoothing Hypergraph Laplacian

The key idea is a central-smoothing assumption: each hyperedge is called *central smooth* if a weighted version of the midpoint of drug node representations is close enough to the representation of the side-effect node. It is motivated by biological research that a side effect of a pair of drugs is caused by a combination of properties of the two drugs [7]. Assuming that representations reflecting all properties of drugs are obtained in a k-dimensional space, the combination containing properties of the two drugs should reflect the corresponding side effects. We show that among commonly used combination operators: average, concatenation, max pooling, and min pooling, the average (also the midpoint) is a good option. First, our operator for combining two drug properties for side effects needs to satisfy the following two criteria: 1) order invariance in the k-dimensional space, since the drug pair has no order and 2) effects of both positive and negative embedding values must be kept to cover the whole embedding space. We can see that the following hold: 1) concatenation violates and 2) max pooling and min pooling violate, but the average (midpoint) satisfies both criteria. In addition, a weighted midpoint, which in the ideal case, would contain properties from each drug and represents a specific combination of the properties, potentially reflecting the cause of a side effect.

1) Central-Smoothing Measure on a Hyperedge: In the embedding space of K-dimension, considering dimension k with the embedding of nodes: $X_k \in \mathbb{R}^{|V|}$ that $X_{k,u} \in \mathbb{R}$ is the embedding of node $u \in V$. Given a hyperedge $e = (u, v, t)$, a weight $W_{k,t} \in \mathbb{R}^+$ is a parameter indicating the relevance of side effect t on dimension k. We assign the weight of side effect t to the hyperedge ($w_k(e) = W_{k,t}$), and let $\mathbf{W}_k = \text{diag}(w_k(e_1), \ldots, w_k(e_{|E|}))$ be the diagonal matrix of the hyperedge weights. The central-smoothing measure on dimension k of the hyperedge is defined as follows:

$$\text{ss}^c(X_k, e) = W_{k,t}\left(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t}\right)^2. \tag{4}$$

2) Central-Smoothing Measure on the Hypergraph: For hypergraph G, the central-smoothing measure on dimension k is defined as the sum of the central-smoothing measures on all hyperedges

$$\text{sh}^c(X_k, G) = \sum_{e\in E} W_{k,t}\left(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t}\right)^2. \tag{5}$$

3) Central-Smoothing Hypergraph Laplacian: Since $\text{sh}^c(X_k, G)$ is a nonnegative quadratic form, there exists an $\mathbf{L}_k \in \mathbb{R}^{|\mathbb{V}|\times|\mathbb{V}|}$, such that $\text{sh}^c(X_k, G) = X_k^T \mathbf{L}_k X_k$. We call $\mathbf{L}_k$ as the *central-smoothing hypergraph Laplacian*, which can be derived as follows.

Let $H \in \mathbb{R}^{|V|\times|E|}$ be a weighted oriented incidence matrix of G that for a hyperedge $e \in E$, $H_{u,e} = H_{v,e} = 1/2$, and $H_{t,e} = -1$, we have

$$\begin{aligned}\text{sh}^c(X_k, G) &= \sum_{e\in E} W_{k,t}\left(\frac{X_{k,u} + X_{k,v}}{2} - X_{k,t}\right)^2 \\ &= X_k^T H \mathbf{W}_k H^\mathsf{T} X_k \stackrel{\text{def}}{=} X_k^T \mathbf{L}_k X_k.\end{aligned} \tag{6}$$

Then

$$\mathbf{L}_k = H\mathbf{W}_k H^\mathsf{T}. \tag{7}$$

4) Computing the Central-Smoothing Hypergraph Laplacian: The central-smoothing hypergraph Laplacian $\mathbf{L}_k$ in (7) can be computed with the time complexity of $O(|E|)$. Concretely, each element $\mathbf{L}_{k,i,j}$ can be computed by

$$\mathbf{L}_{k,i,j} = \sum_{e\in E|i,j\in e} w_k(e) H_{i,e} H_{j,e}. \tag{8}$$

We have four cases as follows.

1) $\mathbf{L}_{k,i,j} = \mathbf{L}_{k,j,i} = 1/4 \sum_{t\in V_s|(i,j,t)\in E} W_{k,t}$ if $i! = j \in V_D$.
2) $\mathbf{L}_{k,i,j} = \mathbf{L}_{k,j,i} = -1/2 n_d(i, j) W_{k,j}$ if $i \in V_D$, $j \in V_S$.
3) $\mathbf{L}_{k,i,i} = 1/4 \sum_{t|t\in V_S} m_d(i, t) W_{k,t}$ if $i \in V_D$.
4) $\mathbf{L}_{k,i,i} = q(i) W_{k,i}$ if $i \in V_S$.

where $n_d(i, j) = |\{(u, v, j) \in E | u = i \vee v = i\}|$, $m_d(i, t) = |\{u|(i, u, t) \vee (u, i, t) \in E\}|$, and $q(i) = |\{(u, v, i)|(u, v, i) \in E\}|$.

a) Complexity analysis: Given N convolution layers, the computational complexity for all central-smoothing hypergraph Laplacian is $O(N \cdot K \cdot |E|)$. Each $\mathbf{L}_k$ can be computed with a complexity of $O(|E|)$ by iterating over all hyperedges in E once, and for each hyperedge, the side-effect weight is added to the corresponding elements in $\mathbf{L}_k$, and we have $N \cdot K$ Laplacian matrices to compute. We note that K here is referred to the size of latent features, and this is not the original input features. In practice, even if the size of the original input features is very large, the number of latent features can be very small ($\leq$200), which is computationally tractable.

5) Non-Weighted Version: In our experiments, we will examine the need for the weight of each side effect. So, we here show a non-weighted version of central-smoothing hypergraph Laplacian, called CentSimple, by fixing $\mathbf{W}_k$ to be an identity matrix, where the central-smoothing hypergraph Laplacian in (7) becomes $\tilde{\mathbf{L}}_k = HH^\mathsf{T}$.

C. Central-Smoothing HGNNs

1) Transforming Input Features to Latent Spaces: We first transform the input feature vector of drugs and one-hot vector of side effects to the K-dimension latent space by using a two-layer feedforward neural network for drugs, and a one-layer feedforward neural network (as an embedding table) for side effect, respectively, as follows:

$$X_D^{(0)} = f_D(X_D);\ X_S^{(0)} = f_S(X_S)$$

where $X_D \in \mathbb{R}^{|K_0|\times|V_D|}$ is the drug input features with feature size K_0, $X_S \in \mathbb{R}^{|V_S|\times|V_S|}$ is the one-hot vector of side effect, $X_D^{(0)} \in \mathbb{R}^{K\times|V_D|}$, $X_S^{(0)} \in \mathbb{R}^{K\times|V_S|}$, and f_D and f_S are the corresponding feedforward neural networks.

2) Convolution Layers on the Latent Spaces: We adapt HGNN layers [5] using $\mathbf{L}_k$ at dimension k. Given hypergraph Laplacian $\mathbf{L}_k$, we have the normalized adjacency matrix with a self-loop at each node: $\tilde{A}_k = 2I - d_{\mathbf{L}_k}^{-1/2} \mathbf{L}_k d_{\mathbf{L}_k}^{-1/2}$, where $d_{\mathbf{L}_k}$ is the degree matrix, corresponding to Laplacian $\mathbf{L}_k$ and I is the identity matrix.

Let $\tilde{D}_k$ be the corresponding degree matrix of $\tilde{A}_k$, and each layer of central-smoothing HGNNs has the following form:

$$X^{(l+1)} = \sigma\left(\tilde{X}^{(l+1)}\Theta^{(l)}\right) \tag{9}$$

where $\tilde{X}^{(l+1)} = [\tilde{x}_1^{(l+1)}, \ldots, \tilde{x}_K^{(l+1)}]$ and $\tilde{x}_k^{(l+1)} = \tilde{D}_k^{-1/2} \tilde{A}_k \tilde{D}_k^{-1/2} x_k^{(l)}$, $\Theta^{(l)} \in \mathbb{R}^{K\times K}$ is the parameters for the transformation from layer (l) to layer $(l + 1)$, and σ is an activation function.

D. Predicting DDIs

Assuming that $X^{*\mathsf{T}} \in \mathbb{R}^{|V|\times K}$ is the final node representation with learned weights $W^* = \{W_k^* | k = 1 \ldots K\}$. For all $e = (u, v, t)$, t is predicted to be a side effect of u and v if the representation of t is close enough to the weighted midpoint of the two drug node representations (computed by score function $p(e, X^*, W^*)$). First, we compute smoothness measures $\text{ssa}(e, X^*, W^*)$ of (u, v, t) on all dimensions

$$\text{ssa}(e, X^*, W^*) = \sum_{k=1}^{K} W_{k,t}^*\left(\frac{X_{k,u}^* + X_{k,v}^*}{2} - X_{k,t}^*\right)^2. \tag{10}$$

Then, the prediction score is defined to be

$$p(e, X^*, W^*) = \frac{1}{1 + \text{ssa}(e, X^*, W^*)}. \tag{11}$$

If $p(e, X^*, W^*) > h$, a predefined threshold, then t is predicted to be a side effect of u and v.

E. Objective Function of CentSmoothie

Let $\bar{E} = V_D \times V_D \times V_S \setminus E$ be complement of the hyperedge set. The objective function to train CentSmoothie is to maximize the score $p(e, X^*, W^*)$ of the known hyperedges and minimize the score of the complement set $\bar{E}^*$. Then, the objective function can be defined as follows:

$$\min_{W^*\geq 0, X^*} f(X^*, W^*) = \sum_{e\in E}(1 - p(e, X^*, W^*))^2 \tag{12}$$

$$+ \lambda \sum_{e\in\bar{E}} p(e, X^*, W^*)^2 \tag{13}$$

where λ is a hyperparameter.

In practice, as $|\bar{E}|$ is too large, we randomly sample a subset of $\Omega \subset \bar{E}, |\Omega| = |E|$ to replace $\bar{E}$ in the objective function to reduce the computational cost (a CentSmoothie implementation available at https://github.com/anhnda/CentSmoothieCode). To keep the non-negative constraint on W^*, we used a projected gradient descent [14].

V. Experiments

We conducted experiments to evaluate the performance of our proposed method, CentSmoothie, a HGNN with a central-smoothing assumption, in two scenarios: 1) a synthetic dataset and 2) three real DDI datasets. On the synthetic dataset, we aimed to validate that CentSmoothie could achieve higher performances than traditional HGNNs, by using the data generated from the central-smoothing assumption. On the real DDI datasets, we examined the performance of CentSmoothie in comparison with baseline models, to prove that the central-smoothing assumption is suitable for DDI data.

For both scenarios, we used 20-fold cross validation using the mean area under the ROC curve (AUC) and the mean area under the precision–recall curve (AUPR) with standard deviations, to validate the prediction performances [1].

For graph and HGNNs, the numbers of layers and the embedding sizes were in [1, 2, 3] and [10, 20, 30], respectively. The activation function was rectified linear unit (ReLu). The hyperparameter λ was fixed: 0.01. The results obtained were the highest performances with the number of layers of 2 and the embedding size of 20 for all methods. All experiments were run on a computer with Intel Core I7-9700 CPU, 8-GB GeForce RTX 2080 GPU, and 32-GB RAM.

A. Synthetic Data

1) Generation: We generated a synthetic dataset with the idea that each drug has several groups of features, and the combination of two groups of features leads to a side effect of the drugs. We fixed the number of drugs $D = 500$, the number of side effects: $S = 45$, and changed the maximum number of groups of drug features from 1 to 6. The detail of the generation process can be found in the Supplementary Material.

2) Comparing Methods: For the synthetic dataset, we used the central-smoothing HGNNs CentSmoothie, the non-weighted central-smoothing HGNNs CentSimple, and the existing spectral-based HGNN, pairwise smoothing hypergraph neural network (HPNN) [5].

3) Results: Fig. 2 shows the AUC and AUPR of each compared method, obtained by changing the maximum number of groups of features for drugs. We could easily see that CentSmoothie achieved the highest AUC and AUPR scores for all values of x-axis, followed by CentSimple and then HPNN. In particular, the AUC scores of CentSmoothie were always higher than 0.95, while those of HPNN decreased when drugs are more complex with larger numbers of groups drugs features. This clearly showed that CentSmoothie could correctly capture the patterns generated by the central-smoothing assumption, particularly for larger numbers of groups of drug features. Similarly, the AUC scores of CentSimple decreased with higher number of maximum number of groups of features, e.g., around 0.75 at 6. The pattern for AUPR scores was also similar to that of AUC scores. This result showed that CentSmoothie could learn different side effects for drug pairs more effectively than CentSimple, implying the significance of using a weight for each side effect in CentSmoothie.

B. Real Data

1) Data Description: We used three real DDI datasets: TWOSIDES, CADDDI, and JADERDDI. TWOSIDES is a public dataset

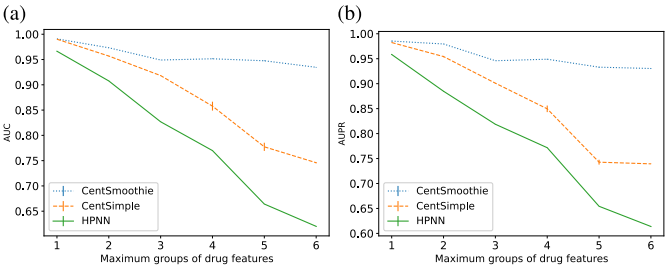

Fig. 2. Synthetic data performance comparison. (a) AUC. (b) AUPR.

for DDI extracted from the FDA adverse event reporting system (U.S. database) [15]. To the best of our knowledge, TWOSIDES is the largest and commonly used benchmark dataset for DDI [1], [8], [16]. In a similar manner as in [15] of TWOSIDES, we used significant tests to generate two new DDI datasets: CADDDI from Canada vigilance adverse reaction report (Canada database, from 1965 to February 2021) [17] and JADERDDI from The Japanese Adverse Drug Event Report (Japanese database, from 2004 to March 2021) [18]. We only selected small molecular drugs appearing in DrugBank [19]. Each drug feature vector was a binary vector with the size of 2329, indicating the existences of 881 substructures and 1448 interacting proteins [20]. The statistics of the final datasets is shown in Table I.

2) Comparing Methods: On the real datasets, we compared our proposed methods to baselines: none-graph-based, graph-based, and hypergraph-based methods. For the none-graph-based method, we used an MLNN [8]. For GNNs, on the drug molecular graphs, we used multiresolution GNN (MRGNN) [16] with the recommended hyperparameter settings. On the DDI graph, we used Decagon [1], a spatial convolution, the spectral convolution graph neural networks (SpecConv) [11], and a heterogeneous GNN (HETGNN) [9]. For HGNN, we used the existing spectral convolution HGNN, HPNN [5]. We also showed the results of CentSimple to see the effect of central smoothing without having weights for side effects.

3) Results: Table II shows the AUC scores and AUPR scores of all methods. We could see that again CentSmoothie achieved the highest AUC and AUPR scores in all three datasets. For TWOSIDES, CentSmoothie achieved 0.9348 in AUC and 0.8749 in AUPR, followed by CentSimple (0.9242 and 0.8638), HPNN (0.9044 and 0.8410), HETGNN (0.9113 and 0.8267), SpecConv (0.8785 and 0.8256), Decagon (0.8639 and 0.8094), MRGNN (0.8452 and 0.8029), and MLNN (0.8372 and 0.7919).

For CADDDI and JADERDDI, CentSmoothie had the highest performances with AUC and AUPR: (0.9845 and 0.8230) and (0.9684 and 0.6044), respectively. The second and third best methods were CentSimple and HPNN, respectively.

In particular, in AUC, there existed two clear performance gaps. The first one was between hypergraph-based methods (CentSmoothie, CentSimple, and HPNN) and non-hypergraph-based methods (HETGNN, SpecConv, Decagon, MRGNN, and MLNN). The second one was between CentSmoothie and (CentSimple and HPNN). The first gap showed the advantage of using the hypergraph-based method for predicting DDI. The second gap showed the advantage of central smoothing over regular smoothing. In addition, we could see the importance of learning weights for each side effect to improve the prediction performance.

In AUPR, there was a clear gap between CentSmoothie and the remaining methods. This again showed the advantage of learning weights under the central-smoothing assumption for predicting DDI.

TABLE I
STATISTICS OF THE THREE REAL DATASETS

Dataset	#drugs	#side effects	#drug-drugs	#drug-drug-side effects	Avg. side effects/drug-drugs	drug-drugs/ side effects		
						Min	*Max*	*Avg*
TWOSIDES	557	964	49,677	3,606,046	72.58	288	22,520	3740.7
CADDDI	587	969	21,918	373,976	17.06	89	3288	385.9
JADERDDI	545	922	36,929	222,081	6.01	60	1922	240.9

TABLE II
COMPARISON OF PERFORMANCES OF THE METHODS ON THE REAL DDI DATASETS

Method	TWOSIDES		CADDDI		JADERDDI	
	AUC	*AUPR*	*AUC*	*AUPR*	*AUC*	*AUPR*
MLNN	0.8372 ± 0.0050	0.7919 ± 0.0041	0.8689 ± 0.0021	0.6927 ± 0.0082	0.8578 ± 0.0015	0.3789 ± 0.0020
MRGNN	0.8452 ± 0.0036	0.8029 ± 0.0039	0.9226 ± 0.0015	0.7113 ± 0.0031	0.9049 ± 0.0009	0.3698 ± 0.0019
Decagon	0.8639 ± 0.0029	0.8094 ± 0.0024	0.9132 ± 0.0014	0.6338 ± 0.0029	0.9099 ± 0.0012	0.4710 ± 0.0027
SpecConv	0.8785 ± 0.0025	0.8256± 0.0022	0.8971 ± 0.0055	0.6640 ± 0.0014	0.8862 ± 0.0025	0.5162 ± 0.0047
HETGNN	0.9113 ± 0.0004	0.8267 ± 0.0005	0.9371 ± 0.0004	0.7974 ± 0.0011	0.8989 ± 0.0007	0.5618 ± 0.0012
HPNN	0.9044 ±0.0003	0.8410 ± 0.0007	0.9495 ± 0.0004	0.7020 ± 0.0018	0.9127 ± 0.0004	0.5198 ± 0.0016
CentSimple	0.9242 ± 0.0003	0.8638 ± 0.0011	0.9584 ± 0.0005	0.6890 ± 0.0016	0.9239 ± 0.0007	0.5349 ± 0.0021
CentSmoothie	**0.9348** ± 0.0002	**0.8749** ± 0.0013	**0.9846** ± 0.0001	**0.8230** ± 0.0019	**0.9684** ± 0.0004	**0.6044** ± 0.0025

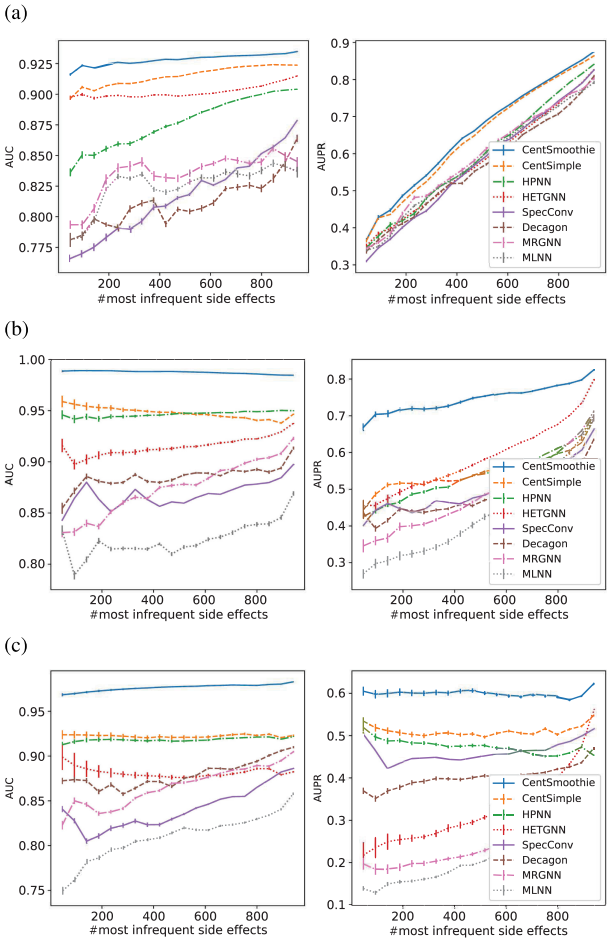

Fig. 3. Performance comparison [AUC (left) and AUPR (right)] on (a) TWOSIDES, (b) CADDDI, and (c) JADDERDDI.

CentSmoothie can learn the representations of side effects together with drugs to leverage the relationships of side effects (see the Supplementary Material for representation visualization of side effects). These side-effect representations might be useful for infrequent side effects, which are harder to predict due to the scarcity of positive training data. Fig. 3 shows the AUC (left) and AUPR (right) scores of the methods on the subset of most infrequent side effects, obtained by starting with the most infrequent side effect and adding the next infrequent side effects to the subset. From both AUC and AUPR scores in Fig. 3, we could see that CentSmoothie achieved the best performances for all values of x-axis (the rightmost point of x-axis corresponds to using all side effects), being followed by CentSimple and HPNN.

VI. CONCLUSION

We have presented CentSmoothie, an HGNN, for predicting DDIs, to learn representations of side effects together with drug representations in the same space. A unique feature of CentSmoothie is a new central-smoothing formulation, which can be incorporated into the hypergraph Laplacian, to model DDIs. Our extensive experiments using both synthetic and three real datasets confirmed clear performance advantages of CentSmoothie over existing hypergraph and GNN methods, indicating that CentSmoothie could learn representations of drugs and side effects simultaneously with the central-smoothing assumption. Furthermore, CentSmoothie kept high performance on the infrequent side effects for which the performances of other methods dropped significantly, indicating that CentSmoothie allows leveraging the relationships among side effects to help the difficult cases of less frequent side effects. For future work, it is interesting to extend the central-smoothing assumption into more general cases not limited to three-uniform hypergraphs. In addition, learning adaptive ratios to replace the constraint of the midpoint might be considered.

REFERENCES

[1] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, Jul. 2018.

[2] Y.-H. Feng, S.-W. Zhang, and J.-Y. Shi, "DPDDI: A deep predictor for drug-drug interactions," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–15, Dec. 2020.

[3] X. Chu, Y. Lin, Y. Wang, L. Wang, J. Wang, and J. Gao, "MLRDA: A multi-task semi-supervised learning framework for drug-drug interaction prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4518–4524.

[4] H. C. Nguyen and H. Mamitsuka, "Learning on hypergraphs with sparsity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2710–2722, Feb. 2020.

[5] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3558–3565.

[6] S. Bai, F. Zhang, and P. H. S. Torr, "Hypergraph convolution and hypergraph attention," 2019, *arXiv:1901.08150*.

[7] K. Corrie and J. G. Hardman, "Mechanisms of drug interactions: Pharmacodynamics and pharmacokinetics," *Anaesthesia Intensive Care Med.*, vol. 12, no. 4, pp. 156–159, Apr. 2011.

[8] N. Rohani and C. Eslahchi, "Drug-drug interaction predicting by neural network using integrated similarity," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, Sep. 2019.

[9] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 793–803.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[13] T.-H.-H. Chan and Z. Liang, "Generalizing the hypergraph Laplacian via a diffusion process with mediators," *Theor. Comput. Sci.*, vol. 806, pp. 416–428, Feb. 2020.

[14] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Aug. 2007.

[15] N. P. Tatonetti, P. P. Ye, R. Daneshjou, and R. B. Altman, "Data-driven prediction of drug effects and interactions," *Sci. Transl. Med.*, vol. 4, no. 125, p. 125, Mar. 2012.

[16] N. Xu, P. Wang, L. Chen, J. Tao, and J. Zhao, "MR-GNN: Multi-resolution and dual graph neural network for predicting structured entity interactions," 2019, *arXiv:1905.09558*.

[17] Canada Vigilance Program. (2021). *Canada Vigilance Adverse Reaction Online Database*. Accessed: May 25, 2021. [Online]. Available: https://www.canada.ca/en/health-canada/services/drugs-health-products/medeffect-canada/adverse-reaction-database.html

[18] Pharmaceutical and Medical Devices Agency. (2021). *The Japanese Adverse Drug Event Report*. Accessed: Mar. 15, 2021. [Online]. Available: https://www.pmda.go.jp/safety/info-services/drugs/adr-info/suspected-adr/0003.html

[19] D. S. Wishart et al., "DrugBank 5.0: A major update to the DrugBank database for 2018," *Nucleic Acids Res.*, vol. 46, no. D1, pp. D1074–D1082, Jan. 2018.

[20] D. A. Nguyen, C. H. Nguyen, and H. Mamitsuka, "A survey on adverse drug reaction studies: Data, tasks and machine learning methods," *Briefings Bioinf.*, vol. 22, no. 1, pp. 164–177, Jan. 2021.