MFSE: A Meta-Fusion Model for Polypharmacy Side-Effect Prediction with Graph Neural Networks

1st Aggelos Ragkousis
Dept. of Electrical and Computer Engineering
University of Patras
Patras, Greece
up1053566@upnet.gr

2nd Vasileios Megalooikonomou
Dept. of Computer Science and Informatics
University of Patras
Patras, Greece
vasilis@ceid.upatras.gr

***Abstract*—Despite being a very popular approach for treating complex diseases, polypharmacy can lead to increased risk of adverse side effects, many of which are observed after the drugs have been released in the market. Luckily, the significant increase in data availability of observed adverse side-effects has paved the way for machine learning approaches to assist in their prediction. In this work, we first present a novel framework for multi-relational link prediction with graph neural networks. Given a multi-relational graph, we create relation-specific vector representations for each node of the graph. With this approach, we create drug vector representations that are side-effect specific, by integrating external molecular and protein-target information with the drug information that is generated directly from the drug-drug interaction prediction graph. With our new meta-fusion approach, each information type is produced from a distinct GNN-based encoder architecture and then the integration is performed according to the side-effect type being predicted. While state-of-the-art models report maximum AUROC scores of 0.91, our technique reaches a score of 0.95. Also, we show that our fusion approach provides valuable external knowledge particularly to drug nodes in the prediction graph that have a smaller node degree.**

I. Introduction

Adverse drug events, described as unintended and undesired effects of medications, have caused serious health threats worldwide. Up to 30% of these adverse drug events are caused by the co-administration of multiple drugs [1], since concomitant drugs can share pharmacological or metabolic pathways. Drug combinations (polypharmacy) are common in therapy, especially for patients with complicated conditions such as cancer [2]. Polypharmacy relies on drug-drug interactions (DDIs), which are modifications of the effect that single drugs cause when administered with other drugs, in order to treat diseases with complex biological processes. However, adverse reactions caused by such modifications lead to nearly 74,000 emergency room visits and 195,000 hospitalizations each year in the US alone [3].

Unfortunately, a large number of DDIs is found by accident after the drugs have been released in the market [3]. The difficulty in their identification can be attributed to the rarity of certain side effects, as well as the high cost and small clinical testing of the experiments [1], [2]. Thus, during the last years, researchers have tried to exploit the computational power of machine learning techniques to predict adverse side effects based on a collection of reported DDIs from scientific sources and adverse drug reaction (ADR) reports [1]. Traditional techniques leverage various chemical and pharmacological information of drugs as features to predict if a drug pair interacts or not [4]–[8]. More recent approaches depict the protein-protein, drug-drug and drug-target interactions (PPI, DTI, DDI) as interconnected large graphs and perform link prediction on the missing edges (side effects) of the DDI graph, without using any chemical information [9]–[11].

In this work, we implement a novel method to integrate chemical characteristics of drugs not only with the drug-drug interaction information deriving from the DDI graph, but also with the protein-target information deriving from the PPI and DTI graphs. To the best of our knowledge, our method is the first to combine all three aspects for the prediction of side effects types. To achieve this, we introduce a new framework to fuse external multi-modal information into a graph where multi-relational link prediction is performed. We use distinct encoders per information type and we combine their output with side-effect specific neural networks. We show that our method outperforms previous approaches, reaching an AUROC score of 0.95. We also demonstrate how the external information particularly assists nodes in sparse regions of the DDI prediction graph. Our method can also be easily extended in other domains where multi-relational link prediction can be used.

II. Previous Work

In this section we present a more detailed review of the most popular machine learning techniques that are used to address the task of adverse side effect prediction for pairs of drugs.

A. Lower Level Algorithms

The *lower*-level algorithms implement a molecular encoder that generates drug vector representations from various chemical and biological characteristics of drugs. The most popular methods rely on the idea of drug similarity, and the assumption that if two drugs 1 and 2 interact to produce a specific biological effect (e.g. a specific side effect), then drugs similar to drug 1 (or drug 2) are likely to interact with drug 1 (or drug 2) to produce the same effect. Thus, different levels of similarity between drugs have been examined, such as chemical sub-structure similarity, target similarity, enzyme similarity and pathway similarity. After the corresponding drug features in each case are encoded into vectors, a metric such as the Tanimoto or the Jaccard coefficient generates the similarity scores. Then, the scores for a pair of drugs are fed into different machine learning models for classification (e.g. logistic regression, random forest, neural network) [5], [7], [8].

Among the examined drug features, the chemical structure information is used in all works described above. It is encoded as a hashed binary vector, where each bit encodes the presence or absence of a substructure in a drug molecule. These vectors are generated using the text-based simplified molecular-input line-entry system (SMILES) [12] and the extended-connectivity fingerprints with a certain diameter, such as diameter 6 (ECFP6) [13]. During the last years, the popularity of Graph Neural Networks (GNNs) [14], [15] paved the way for more powerful molecular encoders [16]–[19]. Molecules can be represented directly as molecule graphs and shape more robust representations than intermediate binary vectors. For each drug pair, the two vector representations are either generated individually and then combined and fed to a final classifier [6], or an inner-message passing is performed jointly in the two molecular graphs [4].

B. Higher Level Algorithms

The *higher*-level algorithms do not directly encode chemical or biological features of drugs or their targeted proteins, but formulate all their interactions as large interaction graphs (DDI, PPI+DTI graphs). Also exploiting the power of Graph Neural Networks, they perform link prediction on the edges (side effects) of the DDI graph. By leveraging the associations of each drug node with its neighbors (drugs and proteins) in the two interaction graphs, the model is able to capture implicit and explicit interaction relationships between drugs and protein-targets that can be related to each specific side-effect type of the drug pair and assist to its prediction. DECAGON [10] and TIP [11] are the most popular algorithms in this category. DECAGON fuses the PPI+DTI and DDI graphs into a single heterogeneous graph. On the contrary, TIP separates the two graphs and implements a cascade architecture, using the vector representations extracted by the protein interactions as input features for the DDI graph in the second stage.

Our model implements a molecular encoder (*lower*) and two interaction graphs (*higher*), and fuses their output information using a side-effect specific meta-fusion strategy.

III. Problem Definition

A. Notations

In this work, we construct three types of graph structures from which we extract important information for the task of side-effect prediction.

1) Node Types: We define the sets of our node types: a) the set of drug nodes $V^d = \{d_1, d_2, \dots, d_{N^d}\}$ with $N^d = |V^d|$ drugs, b) the set of protein nodes $V^p = \{p_1, p_2, \dots, p_{N^p}\}$ with $N^p = |V^p|$ proteins. We also define a subset of the protein set, denoted as $V^t \subset V^p$, which represents the protein nodes that also serve as targets for drug nodes. c) Finally, we define individual sets of atom nodes $V_i^a = \{a_1, a_2, \dots, a_{N_i^a}\}$, one for each drug $d_i \in V^d$, with $N_i^a = |V_i^a|$ atoms.

2) Edge Types: We define the following sets of edge types: a) the set of undirected protein-protein interaction edges $E^{ppi} = \{(p_i, e_{ppi}, p_j) \mid p_i, p_j \in V^p\}$, where (p_i, p_j) represents a pair of proteins and e_{ppi} is the type of their interaction, which is the same for all protein pairs, b) the set of undirected drug-drug interaction edges (side effects) $E^{ddi} = \{(d_i, r, d_j) \mid d_i, d_j \in V^d, r \in R\}$, where (d_i, d_j) represents a pair of drugs and $R = \{r_1, r_2, \dots, r_{N^r}\}$ is the set of different side effect types, with $N^r = |R|$ the number of types. We note that the same pair of drugs (d_i, d_j) can cause multiple side effects $r \in R$. c) the set of directed target to drug interaction edges $E^{dti} = \{(d_i, e_{dti}, p_i) \mid d_i \in V^d, p_i \in V^t\}$, where (d_i, p_i) represents a drug-protein pair and e_{dti} is the type of their interaction directed from p_i to d_i, which is the same for all drug-protein pairs, and d) individual sets of undirected atom-atom molecule interaction edges $E_i^{mol} = \{(a_t, b, a_j) \mid a_t, a_j \in V_i^a, b \in B\}$, one for each drug $d_i \in V^d$, where (a_t, a_j) represents a pair of atoms of drug d_i and $B = \{b_1, b_2, b_3, b_4\}$ is the set of all available bond types between two atom nodes (1: single, 2: double, 3: triple, 4: aromatic). We note that each pair of atom nodes (a_t, a_j) is connected with a single bond type $b \in B$.

3) Graphs: Based on the above definitions, we create two categories of graphs: a) the *prediction graph*, i.e the graph where the task of multi-relational link prediction is performed, which is an undirected drug-drug interaction graph (DDI) denoted as $G^{ddi} = \{V^d, E^{ddi}\}$, and b) the *external graphs*, i.e the graphs that provide valuable information to the *prediction graph*. In particular, we create an undirected protein-protein interaction graph (PPI), denoted as $G^{ppi} = \{V^p, E^{ppi}\}$, followed by a directed drug-target interaction graph (DTI), denoted as $G^{dti} = \{V^{dt}, E^{dti}\}$ where $V^{dt} = V^d \cup V^t$, which jointly provide important protein-target information for the drugs of the DDI network. Also, we create multiple *external* undirected molecule graphs (M) denoted as $G_i^{mol} = \{V_i^a, E_i^{mol}\}$, one for each drug $d_i \in V^d$, which provide molecular information accordingly. The total set of molecule graphs can be defined as $G^{mol} = \{G_1^{mol}, G_2^{mol}, \dots, G_{N^d}^{mol}\}$

B. Multi-relational Link Prediction

We consider the polypharmacy side-effect prediction task as a multi-relational link prediction problem which aims to find the unknown side-effect edges on the drug-drug interaction graph (DDI). More specifically, we assume that we are given only an incomplete subset of known side-effect edges E^{ddi}. Given an edge $(d_i, r, d_j) \notin E^{ddi}$, the task is to assign a probability score p_r^{ij} which determines how likely it is that drugs d_i, d_j are interacting through side-effect type r and that this edge belongs to the complete side-effect set.

For this task, while all *higher* models described above implement an *encoder-decoder* architecture [14], we extend this approach by adding an intermediate *relation-specific encoder*.

More specifically, for the edge triplet (d_i, r, d_j) described above, we use: a) a *node encoder* function $NENC : V^d \rightarrow \mathbb{R}^{d^e}$, which maps drug nodes $d_i, d_j \in V^d$ to drug vector representations $\boldsymbol{h}_i, \boldsymbol{h}_j \in \mathbb{R}^{d^e}$, b) a *relation-specific encoder* function $RENC : \mathbb{R}^{d^e} \rightarrow \mathbb{R}^{d^f}$, which transforms drug vectors $\boldsymbol{h}_i, \boldsymbol{h}_j$ to drug representations of side-effect type r, denoted as $\boldsymbol{z}_{i,r}, \boldsymbol{z}_{j,r} \in \mathbb{R}^{d^f}$, and c) a pairwise *decoder* function $DEC : \mathbb{R}^{d^f} \times \mathbb{R}^{d^f} \rightarrow \mathbb{R}^+$, which assigns to the pair of vectors $(\boldsymbol{z}_{i,r}, \boldsymbol{z}_{j,r})$ an interaction score for side-effect type r.

IV. Modelling

A. Framework of MFSE

In Figure 1, we present the architecture of our model, and more specifically the three main modules described above, namely (a) the *node encoder*, (b) the *relation-specific encoder*, and (c) the *decoder*. More details for each module are given in this section.

B. Node Encoder

Our *node encoder* consists of three distinct encoders shown in the first part (a) of Figure 1, namely the molecular information encoder (M) shown with red colour, which extracts information from an *external* individual molecule graph M for each drug, the drug-drug interaction information encoder (DDI) shown with blue colour, which extracts information from the main DDI *prediction* graph, and the protein-target information encoder (PPI+DTI) shown with yellow colour, which extracts protein-target information from the *external* PPI and DTI graphs. A per-layer update of each encoder type is given in Figure 2.

1) Molecular Information Encoder (M): This encoder aims to create a matrix of drug vector representations based on the information of their molecular structure. The matrix is denoted as $\boldsymbol{H}_d^{mol} \in \mathbb{R}^{N^d \times d^{mol}}$, where N^d is the number of drugs and d^{mol} is the output molecular dimensionality.

Each molecule graph, e.g. graph $G_i^{mol} = \{V_i^a, E_i^{mol}\}$ of drug d_i, has a set of N_i^a atom nodes, with various chemical input features (see section V-A) and total dimensionality $d^a = 69$. Thus, the input matrix for drug d_i is $\boldsymbol{H}_i^{(0)} \in \mathbb{R}^{N_i^a \times d^a}$. The novelty of our molecular encoder lies on the way bond types are modelled. In this work, we model each molecular graph using the R-GCN encoder, as presented in [20]. The R-GCN encoder for an atom node is defined as

$$\boldsymbol{h}_{a_t}^{(l+1)} = ReLU(\sum_{b \in B} \sum_{j \in N^b(t)} \frac{1}{c_{t,b}} \boldsymbol{W}_b^{(l)} \boldsymbol{h}_{a_j}^{(l)} + \boldsymbol{W}_0^{(l)} \boldsymbol{h}_{a_t}^{(l)}) \quad (1)$$

$$\boldsymbol{W}_b^{(l)} = \sum_{k=1}^{K} \boldsymbol{a}_{bk}^{(l)} \boldsymbol{V}_k^{(l)} \quad (2)$$

where $\boldsymbol{h}_{a_t}^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of atom $a_t \in V_i^a$ in the l-th layer of the neural network, $d^{(l)}$ is the correspondent dimensionality and $N^b(t)$ is the set of all first-order atom neighbors of atom node a_t, that are connected to this atom with bond type $b \in B$. Also, $c_{t,b}$ is a normalization constant defined as $c_{t,b} = |N^b(t)|$. We use four different bond types (b_1: single, b_2: double, b_3: triple, b_4: aromatic), where each bond type is associated with a separate weight matrix $\boldsymbol{W}_b$ and bond-specific aggregations are performed. After all vectors have been generated in parallel for all atom nodes of the drug, a global average pooling function generates the final vector representation of the drug:

$$\boldsymbol{h}_{d_i}^{(l+1)} = MEAN(\boldsymbol{h}_{a_t}^{(l+1)} \mid a_t \in V_i{}^a) \quad (3)$$

The weight $\boldsymbol{W}_b^{(l)}$ is defined in Equation 2 as a linear combination of basis transformations $\boldsymbol{V}_k^{(l)} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ with coefficients $\boldsymbol{a}_{bk}^{(l)}$ such that only the coefficients depend on the bond $b \in B$. This method is called basis-decomposition and is suggested by the authors in [20] to address the issue of the rapid growth in the number of parameters due to multiple relation types. Our idea to increase the significance of bond types is based on the assumption that some bond types can be more influential than others in the prediction of different side effects, due to their different chemical properties.

2) Drug-Drug Interaction Information Encoder (DDI): This encoder aims to create a matrix of drug vector representations based on the information of all side effects types that drug nodes could cause when administered concurrently with their drug neighbors. The matrix is denoted as $\boldsymbol{H}_d^{ddi} \in \mathbb{R}^{N^d \times d^{ddi}}$, where d^{ddi} is the output dimensionality. For the drug input features we use one-hot encoding $\boldsymbol{H}_d^{(0)} \in \mathbb{R}^{N_d \times N_d}$. As in the molecular encoder, we model our DDI graph using the R-GCN encoder [20]. For each drug node, the encoder is defined as:

$$\boldsymbol{h}_{d_i}^{(l+1)} = ReLU(\sum_{r \in R} \sum_{j \in N^r(i)} \frac{1}{c_{i,r}} \boldsymbol{W}_r^{(l)} \boldsymbol{h}_{d_j}^{(l)} + \boldsymbol{W}_0^{(l)} \boldsymbol{h}_{d_i}^{(l)}) \quad (4)$$

$$\boldsymbol{W}_r^{(l)} = \sum_{k=1}^{K} \boldsymbol{a}_{rk}^{(l)} \boldsymbol{V}_k^{(l)} \quad (5)$$

where $\boldsymbol{h}_{d_i}^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of drug $d_i \in V_i{}^d$ in the l-th layer of the neural network, $d^{(l)}$ is the correspondent dimensionality and $N^r(i)$ is the set of of all first-order drug neighbors of drug node d_i, that are connected to this drug with side-effect type $r \in R$. Also, $c_{i,r}$ is a normalization constant defined as $c_{i,r} = |N^r(i)|$. Again, the weight $\boldsymbol{W}_r^{(l)}$ is defined in Equation 5 as a linear combination of basis transformations. Indicative aggregations of the DDI encoder are given in part (b) of Figure 2.

3) Protein-Target Information Encoder (PPI+DTI): This encoder aims to create a matrix of drug vector representations based on the information of their protein targets. The matrix is denoted as $\boldsymbol{H}_d^{dti} \in \mathbb{R}^{N^d \times d^{dti}}$, where N^d is the number of drug nodes and d^{dti} is the output dimensionality.

This module consists of the protein-protein interaction encoder (PPI) followed by the drug-target interaction encoder (DTI). The first generates a matrix of protein embeddings $\boldsymbol{H}_p^{ppi} \in \mathbb{R}^{N^p \times d^{ppi}}$, where N^p is the number of protein nodes and d^{ppi} is the output dimensionality of the protein vector representations. For the protein input features we use one-hot

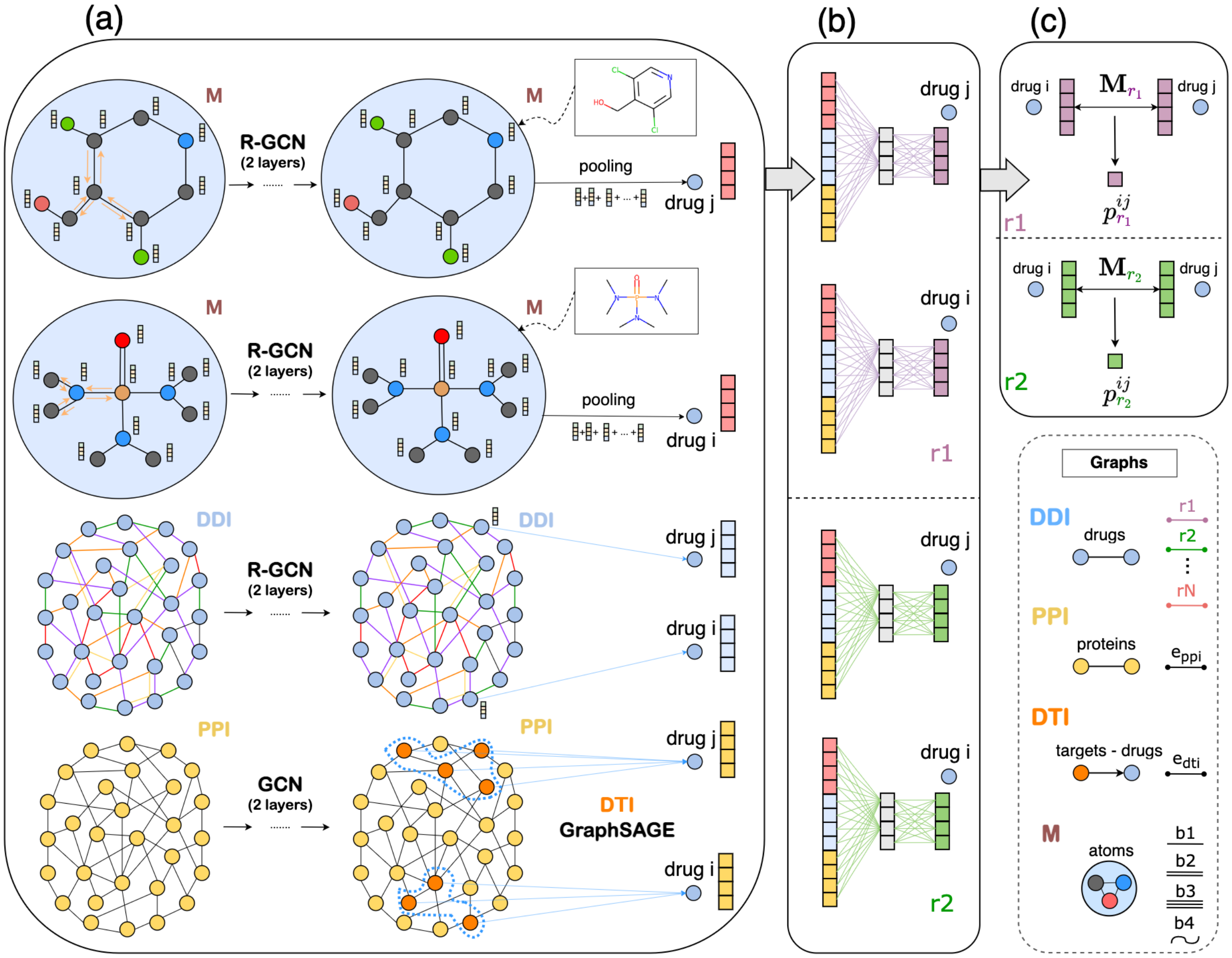

Fig. 1. Overview of the MFSE architecture. (a) A high-level overview of the node encoder for a pair of drugs (d_i, d_j). (b) An overview of the relation-specific encoder for the prediction of two random side effects $r_1(purple)$ and $r_2(green)$ of the drug pair. (c) An overview of the model decoder for the prediction of the two side effects of the drug pair.

encoding $\boldsymbol{H}_p^{(0)} \in \mathbb{R}^{N_p \times N_p}$. We model this graph for each protein node using the GCN module [15], defined as:

$$\boldsymbol{h}_{p_i}^{(l+1)} = ReLU(\frac{1}{c_i} \sum_{j \in N(i)} \boldsymbol{W}_p^{(l)} \boldsymbol{h}_{p_j}^{(l)}) \tag{6}$$

where $\boldsymbol{h}_{p_i}^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of protein $p_i \in V^p$ in the l-th layer of the neural network, and $c_i = |N(i)|$ is the number of all first-order protein neighbors of node p_i.

After all protein vector representations have been generated, the DTI encoder is used to combine the vectors of the proteins that serve as targets to each drug and generate the drug vector representations. For this task, we use a one-layer GraphSAGE module [21] to transform the protein vectors, along with one-hot encoded drug input vectors $\boldsymbol{H}_d \in \mathbb{R}^{N_d \times N_d}$, into drug representations, defined as:

$$\boldsymbol{h}_{d_i}^{dti} = ReLU(\frac{1}{c_i} \sum_{j \in N(i)} \boldsymbol{W}_t \boldsymbol{h}_{p_j} + \boldsymbol{W}_0 \boldsymbol{h}_{d_i}) \tag{7}$$

where $c_i = |N(i)|$ is the number of all first-order protein neigbours (targets) of drug node $d_i \in V^d$. Similarly with the molecular encoder, an example of the aggregations performed in the (PPI+DTI) stage is given in part (c) of Figure 2.

C. Relation-Specific Encoder

After the molecular, protein-target and drug-drug interaction vector representations have been generated in parallel for all drugs from individual encoders (which jointly represent the *node encoder* of the model), the *relation-specific encoder* shown in part (b) of Figure 1 fuses them into side-effect specific drug representations. Inspired by the method of binary relevance [22] for multi-label classification, we implement a meta-fusion scheme consisting of a distinct neural network per relation type, in order to treat each type as a separate binary classification problem and to exploit different associations between the three distinct sources of drug information. Our fusion approach presents a novel way to fuse valuable *external* information to the main *prediction* DDI graph.

More formally, we use m layers $(l_1, \ldots, l_m)$ of $n = N^r$ parallel neural networks in order to produce n vector representations for each drug (one for each side-effect type), according

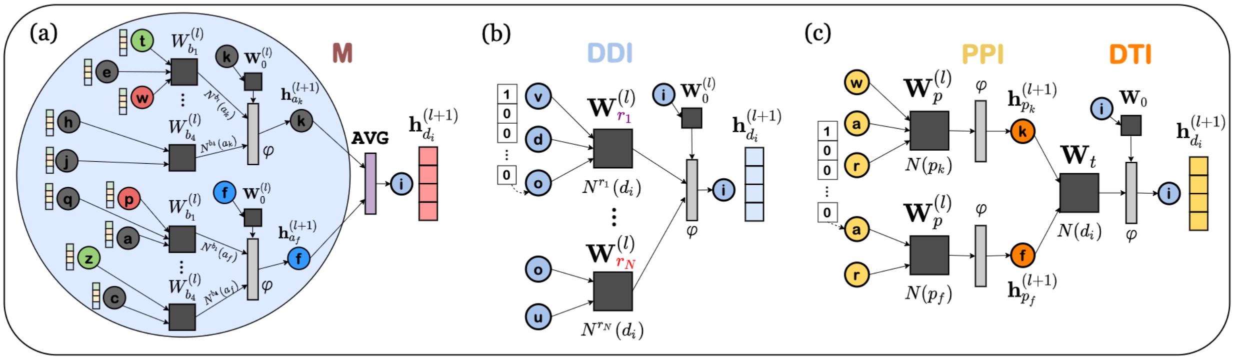

Fig. 2. A detailed explanation of a per-layer update of the model encoders for a single drug node d_i: (a) molecular encoder (M), (b) drug-drug interaction information encoder (DDI), (c) protein-target information encoder (PPI+DTI).

to the following equations:

$$\boldsymbol{X}^{(0)} = \boldsymbol{H}_d = CONCAT(\boldsymbol{H}_d^m, \boldsymbol{H}_d^{dti}, \boldsymbol{H}_d^{ddi}) \tag{8}$$

$$l_1 : \begin{cases} \boldsymbol{X}_{r_1}^{(1)} = RELU(\boldsymbol{W}_{r_1}^{(0)} \boldsymbol{X}^{(0)} + \boldsymbol{b}_{r_1}^{(0)}) \\ \quad \dots \\ \boldsymbol{X}_{r_n}^{(1)} = RELU(\boldsymbol{W}_{r_n}^{(0)} \boldsymbol{X}^{(0)} + \boldsymbol{b}_{r_n}^{(0)}) \end{cases} \tag{9}$$

$$\dots$$

$$l_m : \begin{cases} \boldsymbol{Z}_{r_1} = \boldsymbol{X}_{r_1}^{(m)} = \boldsymbol{W}_{r_1}^{(m-1)} \boldsymbol{X}_{r_1}^{(m-1)} + \boldsymbol{b}_{r_1}^{(m-1)} \\ \quad \dots \\ \boldsymbol{Z}_{r_n} = \boldsymbol{X}_{r_n}^{(m)} = \boldsymbol{W}_{r_n}^{(m-1)} \boldsymbol{X}_{r_n}^{(m-1)} + \boldsymbol{b}_{r_n}^{(m-1)} \end{cases} \tag{10}$$

We denote as $\boldsymbol{X}^{(0)} \in \mathbb{R}^{N^d \times d^e}$ in Equation 8 the shared, concatenated input vector with $d^e = d^{mol} + d^{dti} + d^{ddi}$ that is fed to the first layer l_1 in all N^r networks. In Equations 9 and 10, we denote as $\boldsymbol{X}_r^{(1)} \in \mathbb{R}^{N^d \times d^{(1)}}$ and $\boldsymbol{Z}_r \in \mathbb{R}^{N^d \times d^f}$ the output drug matrices of the neural networks corresponding to each side effect type $r \in R$ for the first and last layers l_1 and l_m of the meta-fusion level accordingly, where $d^{(1)}$ and $d^{(m)} = d^f$ are the drug output dimensionalities of the two levels respectively. After the first layer, the side-effect specific output $\boldsymbol{X}_r$ of each network serves as input to that network in the next layer. We denote the full output matrix of side-effect vector representations of the meta-fusion level as $\boldsymbol{Z} \in \mathbb{R}^{N^r \times N^d \times d^f}$. In order to reduce the dimensionality of matrices $\boldsymbol{W}_r^{(0)} \dots \boldsymbol{W}_r^{(m-1)}$ for each side-effect type $r \in R$, we again define them as a linear combination of basis transformations, as in Equations 2 and 5.

It is important to point out that the input drug vector representations generated from our *node encoder* are the same for all side-effect specific neural networks of our *relation-specific encoder*. Due to the fact that the individual networks are trained in parallel and share the same *node encoder* as input, our *node encoder* adjusts its learnable parameters based on all side-effect types and thus the individual networks influence each other. This information exchange assists our binary classifiers to generalize better and prevents our model from over-fitting.

D. Decoder

The *decoder* of our model is shown in part (c) of Figure 1. It receives as input the drug representations from the previous level and calculates the probability p_r^{ij} that a drug pair (d_i, d_j) will cause side-effect r. In our work, we use the DistMult Factorization decoder [23], which was also used in [10], [11] for the task of multi-relational link-prediction. However, in our case, the generated drug representations of the drug pair are side-effect specific vectors $(\boldsymbol{z}_{i,r}, \boldsymbol{z}_{j,r}) \in \boldsymbol{Z}$, and the probability is calculated as:

$$p_r^{ij} = \sigma({\boldsymbol{z}_{i,r}}^T \boldsymbol{M}_r \boldsymbol{z}_{j,r}) \tag{11}$$

where $\boldsymbol{M}_r$ is a trainable diagonal matrix associated with r.

E. Model Training

The model is trained in an end-to-end fashion using the cross-entropy loss:

$$L_r^{ij} = -\log{(p_r^{ij})} - \mathbb{E}_{m \sim P_j^r} \log{(1 - p_r^{im})} \tag{12}$$

We use the following negative sampling strategy. For each edge triplet $(d_i, r, d_j) \in E^{ddi}$, we sample two random drug nodes $d_{n1}, d_{n2} \in V^d$ and we form a negative edge of that side-effect type (d_{n1}, r, d_{n2}). The final loss is given by the sum of all losses:

$$L = \sum_{(d_i, r, d_j) \in E} L_r^{ij} \tag{13}$$

V. Experiments

In this section, we provide more details about our dataset and we demonstrate the effectiveness of MFSE compared to previous works.

A. Dataset

1) Interaction Graphs: We use the BioSNAP-Decagon [24] dataset which was introduced in [10]. For the PPI graph, the dataset consists of 19,081 protein nodes and 715,612 protein-protein interaction edges. The 3,648 target nodes are a subset of these proteins and interact with a set of 645 drugs in the DTI graph, which has 18,596 drug-target interaction edges. Finally, the DDI graph consists of 4,625,608 total side-effect edges of 1,097 different types. Following the preprocessing of [10], only the side effect types that occurred in at least 500 drug pair combinations are used in the dataset.

2) Molecule Graphs: In order to extract the graph molecular structure of each drug, we mapped the PubChem [25] IDs of drugs provided in the BioSNAP dataset with their corresponding DrugBank [26] IDs and we obtained their SMILES string. Then, we used TorchDrug, a machine learning platform designed for drug discovery [27], to convert each SMILES string to a molecule graph with node features and bond-edge types.

The atom features extracted by the TorchDrug library are the atomic symbol, the atomic chiral tag, the degree of the atom in the molecule (including Hs), the number of formal charges in the molecule, the total number of Hs (explicit and implicit) on the atom, the number of radical electrons on the atom, the atom's hybridization, whether the atom is aromatic, whether the atom is in a ring and the 3D position of the atom. All these features are one-hot encoded.

B. Settings

Regarding the *node encoder*, we use two R-GCN layers for the molecular information encoder (M), where $d^{(0)} = d^{(1)} = d^{mol} = 32$. For the target information encoder (PPI+DTI), we use two GCN layers for the PPI network with $d^{(0)} = 64$ and $d^{ppi} = d^{(1)} = 32$ accordingly, and a single GraphSAGE layer for the DTI network with $d^{dti} = d^{(0)} = 32$. Lastly, we use two R-GCN layers for the drug-drug interaction information encoder (DDI), where $d^{(0)} = 64$ and $d^{ddi} = d^{(1)} = 32$ respectively. Thus, the concatenated output size of the model encoder for each drug is $d^e = 96$, with equal contribution of each information type $d^{mol} = d^{dti} = d^{ddi} = 32$. Lastly, for the *relation-specific encoder*, we use two layers with $d^{(0)} = 32$ and $d^f = d^{(1)} = 32$. For all R-GCN and meta-fusion layers we use $K = 32$ for basis decomposition.

We use standard 10-fold cross validation, so that the edge sets of all side-effect types randomly split into 10 non-overlapping folds and each fold is given an opportunity to be used as a test set, whilst all other folds collectively are used as the training set (9:1 ratio for each test fold of each side-effect type). We train the model for 200 epochs in full-batch configuration, i.e. the whole dataset is fed into the model in each epoch. Our model is implemented in Pytorch using the Pytorch Geometric package [28]. Experiments are performed on a Tesla V100-PCIE-32GB GPU.

The following metrics are used to measure the performance of our model: 1) AUROC score: area under the receiver-operating characteristic and 2) AUPRC: area under precision-recall curve.

C. Baselines and Ablation Study

We compare our implementation with the recent *higher*-level models discussed in section II, namely DECAGON [10] and TIP [11], because they outperform the *lower*-level models and both use the DDI and PPI-DTI information, as our model.

In order to assess the performance of our fusion scheme, we also implement a different version of our model (MFSE-LF) shown in part (b) of Figure 3, where our side-effect specific meta-fusion level has been removed and a late-fusion stage has been added after the DistMult decoder. Moreover, we test the performance of our three encoders individually, without any fusion level, in order to evaluate their contribution and to compare them with the joint multi-modal network. MFSE-M, MFSE-DDI and MFSE-PPI-DTI use only the drug-drug interaction, molecular and protein-target information encoders respectively and are presented in parts (c)-(e) of Figure 3.

VI. Results

A. Performance Comparison

In Table I, we report the results of our baseline models (the results are given as stated by their authors and were successfully reproduced). We observe that our model (MFSE) presents an improvement of 9% in AUROC compared to DECAGON and 4% compared to TIP. The increase in AUPRC score is 12.5% and 5.2% respectively over the two models.

Additionally, in Table II, we report the results of our ablation study, in the form of mean and standard deviation after performing 10-fold cross validation. Firstly, we observe that all individual encoders present significant AUROC and AUPRC scores, suggesting that they can provide meaningful insights to our task. It is also remarkable that our MFSE-PPI-DTI model, which only uses *external* protein-target information, can generate meaningful vector representations for the drugs of the DDI graph and presents comparable performance with MFSE-DDI. Additionally, we observe that our proposed meta-fusion MFSE model outperforms all individual encoders, and in section VI-B it will be shown that this improvement particularly favors drug nodes with limited side-effect information, which is a crucial fact for real-life applications. We also observe that the MFSE-LF version fuses the information sources less efficiently, while also being computationally much more expensive because it uses three DistMult decoders instead of one, as can be seen from Figure 3. In part (a) of Figure 4, we plot the increase of AUPRC score with the number of epochs for a random fold for all MFSE versions of the ablation study.

TABLE I
Performance Comparison with Baselines

Model	Graph Types	AUPRC	AUROC
DECAGON	PPI+DTI, DDI	0.832	0.872
TIP	PPI+DTI, DDI	0.890	0.914
MFSE	M, PPI+DTI, DDI	**0.936**	**0.951**

B. The Effectiveness of our Fusion Strategy

The main focus of our approach is to effectively fuse *external* information to the DDI *prediction* graph. According to the results of the authors in [11] (which were also verified by us), TIP exhibits a negligible improvement of 0.66% in AUROC compared to its respective version that relies only on the DDI

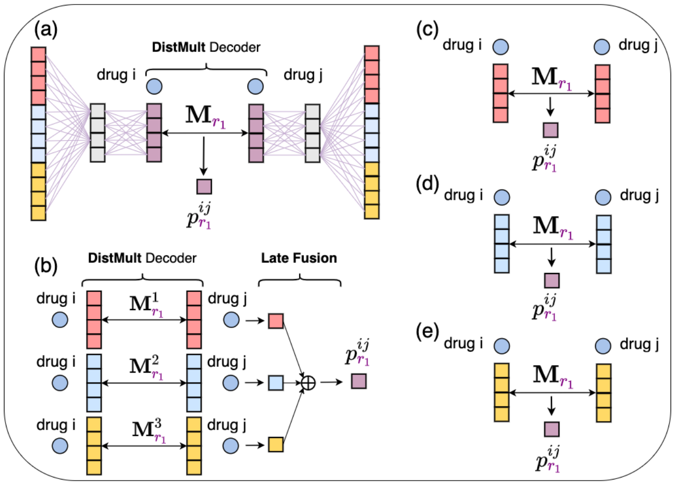

Fig. 3. A comparison of different fusion approaches. (a) MFSE: relation-specific encoder with meta-fusion (our proposed method). (b) MFSE-LF: late-fusion. (c) MFSE-M: no fusion, only molecular information used. (d) MFSE-DDI: no fusion, only drug-drug interaction information used. (e) MFSE-PPI-DTI: no fusion, only protein-target information used.

TABLE II
ABLATION STUDY

Model	Graph Types	AUPRC	AUROC
MFSE	M, PPI+DTI, DDI	**0.936±0.001**	**0.951±0.001**
MFSE-M	M	0.850±0.004	0.868±0.004
MFSE-DDI	DDI	0.923±0.001	0.940±0.001
MFSE-PPI-DTI	PPI+DTI	0.925±0.001	0.941±0.001
MFSE-LF	M, PPI+DTI, DDI	0.929±0.001	0.945±0.001

prediction graph and does not use any protein-target information. MFSE aims to increase the influence of PPI-DTI *external* graphs, while also integrating valuable information from the M graphs. While TIP connects the PPI-DTI graph as input to the DDI graph, we use our novel *relation-specific encoder* with meta-fusion to combine all information sources in the end with side-effect specific neural networks. Comparing our MFSE model with our respective version without any *external* protein-target and molecular information, named MFSE-DDI, we indeed record an improvement of 1.2% in AUROC and 1.4% in AUPRC, which is higher than that of TIP mentioned above. But, most importantly, in the rest of this section we will show that the *external* information particularly assists drug pairs for which we possess less information from the DDI graph. This is extremely important for real-life applications, because, unlike the existing side-effect information which is incomplete, the molecular and protein-target information is always available for all drugs.

By looking at the dataset, we observe that the distribution of side-effect edges is highly imbalanced between the different side-effect types. However, in total, the DDI graph is a very dense network, with 9,251,216 undirected positive side-effect edges. Thus, due to the DDI-encoder, drug nodes that may have few or no edges of a specific side-effect type with other drugs can still use edges of other side-effect types that they form with their neighbours in order to infer implicit information about this type. This sharing of information allows drug nodes to generalize to unseen side-effect edges of rare types, despite having inadequate training edges of these types. To prove that, we divide our side-effects types into 5 equal-sized bins based on their number of (undirected) edges and we average their AUROC scores from our MFSE-DDI model (Figure 4, part (b)). It is evident that all side-effect bins present comparable AUROC scores, irrespective of the number of edges, and no side-effect bin has a score less than 0.94. However, if we evaluate the model per drug and not per side-effect type, one main question can be raised. What happens to drug nodes whose total node degree is small, i.e. their total number of associated edges of all side-effect types is much less compared to other nodes? We expect that, on average, drugs that belong to sparse regions of the network should under-perform because they lack both explicit information about certain side-effect types and implicit information deriving from other types. Based on this assumption, we expect our multi-modal MFSE model to provide significant molecular and protein-target information especially to these drugs in order to overcome the limited information of the DDI network.

To investigate this issue, we group the 645 drug nodes to 3 equal sized bins of 215 drugs according to their total node degree of training edges, including both positive and negative edges after negative sampling is performed (since the number of training negative edges also affects the learning ability of sparse nodes). We evaluate the AUPRC score for each drug bin and for each side-effect type, and we calculate the total AUPRC score for each bin by averaging the scores of all side-effect types, as shown in part (c) of Figure 4. Comparing MFSE-DDI with our full MFSE suggested model, we observe that our approach presents a significant improvement of 17% in AUPRC for the first bin of nodes (nodes with smaller node degrees), and less improvement as the node degree increases and the DDI network possesses significant information to make accurate predictions in dense regions of the graph.

VII. CONCLUSIONS AND FUTURE WORK

In this work, we propose MFSE, a meta-fusion model for combining multi-modal information effectively for the task of polypharmacy side-effect prediction. To the best of our knowledge, our approach is the first that fuses three important information sources (molecular, target and drug-drug interaction information) and outperforms previous state-of-the-art approaches, reaching an AUROC score of 0.95 and an AUPRC score of 0.93. To achieve that, we develop a novel framework for multi-relational link prediction in graph neural networks, which generates relation-specific node vector representations. More specifically, we extract information from separate graph neural network encoders optimized for each type, and then we integrate the external drug information sources with the information from the main prediction graph in a meta-fusion fashion in order to generate meaningful drug side-effect embeddings. We evaluate our model encoders individually to depict their effectiveness and we show that our fusion scheme yields better results compared to a more

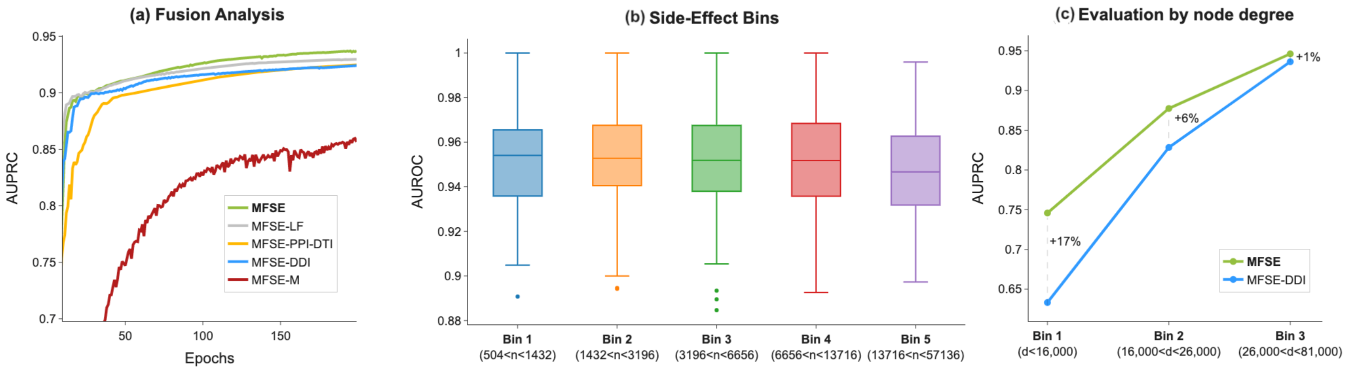

Fig. 4. Evaluation plots: (a) ablation study, (b) Side-Effect Bins based on their number of undirected edges n, (c) evaluation by node degree d.

traditional late-fusion strategy. More importantly, we show that our model particularly assists drug nodes that have less available side-effect edges with other drugs, by leveraging valuable molecular and target information.

Regarding our future work, there are several directions for extensions. While our meta-fusion level concatenates information from distinct GNN-based encoders that are problem-specific, it could also integrate additional drug features to the main *prediction* graph, which could either be directly concatenated at the fusion stage or generated from other encoder types such as more traditional neural network architectures. Also, the 3-stage architecture of MFSE, consisting of the two encoders and the decoder, could be used as a general framework for multi-relational link prediction problems even in non-biomedical domains, where valuable external information could be fused in the *prediction* graph.

References

[1] N. P. Tatonetti, G. H. Fernald, and R. B. Altman, "A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports," *Journal of the American Medical Informatics Association*, vol. 19, pp. 79–85, 2012.

[2] M. Bansal, J. Yang, C. Karan, M. P. Menden, J. C. Costello, H. Tang, G. Xiao, Y. Li, J. Allen, and R. Zhong, "A community computational challenge to predict the activity of pairs of compounds," *Nature biotechnology*, vol. 32, pp. 1213–1222, 2014.

[3] B. Percha and R. B. Altman, "Informatics confronts drug–drug interactions," *Trends in pharmacological sciences*, vol. 34, pp. 178–184, 2013.

[4] A. Deac, Y.-H. Huang, P. Veličković, P. Liò, and J. Tang, "Drug-drug adverse effect prediction with graph co-attention," *arXiv preprint arXiv:1905.00534*, 2019.

[5] J. Y. Ryu, H. U. Kim, and S. Y. Lee, "Deep learning improves prediction of drug–drug and drug–food interactions," *Proceedings of the National Academy of Sciences*, vol. 115, pp. E4304–E4311, 2018.

[6] X. Cao, R. Fan, and W. Zeng, "Deepdrug: a general graph-based deep learning framework for drug relation prediction," *biorxiv*, 2020.

[7] S. Seo, T. Lee, M. hyun Kim, and Y. Yoon, "Prediction of side effects using comprehensive similarity measures," *BioMed Research International*, vol. 2020, 2020.

[8] Y. Deng, X. Xu, Y. Qiu, J. Xia, W. Zhang, and S. Liu, "A multimodal deep learning framework for predicting drug–drug interaction events," *Bioinformatics*, vol. 36, pp. 4316–4322, 2020.

[9] B. Malone, A. García-Durán, and M. Niepert, "Knowledge graph completion to predict polypharmacy side effects." Springer, 2018, pp. 144–149.

[10] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, pp. i457–i466, 2018.

[11] H. Xu, S. Sang, and H. Lu, "Tri-graph information propagation for polypharmacy side effect prediction," *arXiv preprint arXiv:2001.10516*, 2020.

[12] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[13] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, pp. 742–754, 2010.

[14] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

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

[16] N. D. Cao and T. Kipf, "Molgan: An implicit generative model for small molecular graphs," *arXiv preprint arXiv:1805.11973*, 2018.

[17] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," *Advances in neural information processing systems*, vol. 31, 2018.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry." PMLR, 2017, pp. 1263–1272.

[19] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, 2015.

[20] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks." Springer, 2018, pp. 593–607.

[21] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[22] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: an overview," *Frontiers of Computer Science*, vol. 12, pp. 191–202, 2018.

[23] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[24] M. Zitnik, R. Sosič, S. Maheshwari, and J. Leskovec, "BioSNAP Datasets: Stanford biomedical network dataset collection," http://snap.stanford.edu/biodata, aug 2018.

[25] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant, "Pubchem: a public information system for analyzing bioactivities of small molecules," *Nucleic acids research*, vol. 37, pp. W623–W633, 2009.

[26] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali, "Drugbank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic acids research*, vol. 36, pp. D901–D906, 2008.

[27] Z. Zhu, C. Shi, Z. Zhang, S. Liu, M. Xu, X. Yuan, Y. Zhang, J. Chen, H. Cai, and J. Lu, "Torchdrug: A powerful and flexible machine learning platform for drug discovery," *arXiv preprint arXiv:2202.08320*, 2022.

[28] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.