LUISS

Degree Program in Data Science and Management

Data Visualization Course

Graph Neural Networks with Application in Bioinformatics

Prof. Blerina Sinaimeri

SUPERVISOR

Prof. Alessio Martino

CO-SUPERVISOR

Claire Probst 785051

CANDIDATE

Academic Year 2025/2026

Contents

Abstract

Adverse drug reactions (ADR) pose a challenge in drug development and in healthcare due to the range of an unintended response from a given drug, from mild to life-threatening, to patients. Additionally, traditional methods of identifying ADRs are costly and may not portray all possible reactions and side effects. Therefore, the development of a computational method to identify unknown drug-side effect associations utilizing molecular interaction, chemical, and semantic relationships is a beneficial alternative and possible improvement to traditional methods and early identification of ADRs. This thesis aims to construct a bi-adjacency matrix of known and unobserved drugs and side effects and construct a heterogeneous graph by integrating known drug-side effect association network, drug-drug semantic similarity network, drug-drug chemical structure similarity network, and both drug-protein and protein-protein interaction networks. A Relational Graph Convolution Network (R-GCN) encoder and Generalized Matrix Factorization decoder will then be applied to the resulting heterogeneous graph with the goal of predicting previously unobserved drug-side effects from the initial known drug-side effect adjacency matrix through matrix reconstruction.

1 Introduction

An adverse reaction to a given drug as defined by the World Health Organization is a reaction that is noxious, unintended, and occurs at doses normally used in man (2). This definition can be expanded upon to consider possible actions for intervention against the adverse drug reaction (ADR) as "an unpleasant reaction, resulting from an intervention related to the use of a medicinal product, which predicts hazard from future administration and warrants prevention or specific treatment, or alteration of the dose regime, or withdrawal from the product" (3). Regardless, ADRs have the potential to impose health safety risk and concern, ineffective treatment methods, and financial burdens from the aspect of patients. While ADR incidences may be induced by drug interactions or patient risk factors such as age, gender, genetic predisposition, and comorbidities (4), the importance of early ADR detection to mitigate or avoid these risks cannot be understated and is a challenging task that healthcare providers, researchers, and drug manufacturers face.

Traditional ADR detection methods include reporting systems, epidemiological studies, signal detection (4) and laboratory methods (5). Spontaneous reporting systems require a voluntary report of ADRs from healthcare providers and patients to health authorities and pharmaceutical companies for further analysis. While this method is both low-cost and takes advantage of real-world data of how drugs react outside of clinical trials, the issues of underreporting ADRs resulting in an incomplete drug safety profile and reporting bias are present. The methods of preclinical testing and clinical trials occur at the beginning stages of drug development to detect ADRs prior to marketing the drug. However, because not all ADRs may be identified, epidemiological studies are conducted post-marketing. Cohort studies and case-control studies fall under this umbrella, with the intention of investigating drug use and adverse reactions through large population analysis. In comparison to spontaneous reporting, there is a more stable basis to infer and establish the causality between drugs and ADRs, but it may be difficult to completely isolate specific drug effects from other present variables. Additionally, such studies are time- and cost-intensive. The third traditional ADR detection method is signal detection, which identifies ADRs through analysis of ADR databases. An example is the FDA's Adverse Event Reporting System (FAERS) maintained by the Food and Drug Administration of the United States, which supports post-market surveillance of drugs through adverse event reports, medication error reports, and product quality complaints (6) made by patients or physicians. Another similar database is the Side Effect Resource (SIDER) database of drugs and side effects, which contains a limited set of marketed medications and their known ADRs (7). The main advantages of signal detection on databases are scalability and proactivity: algorithms traverse datasets quickly and allow for consistent, updated monitoring (4). The limitation of this method is

lack of data quality, specifically completeness, which may lead to unreliable and incorrect signal detection. The fourth traditional ADR detection method is a laboratory method known as *in vitro* safety pharmacology profiling (SPP), in which biochemical and cellular assays are utilized to assess the potential presence and potency of ADRs in newly discovered drugs at an early phase of development. Though drug development pipelines may be less 'clogged' with low quality drugs, as in those with a high number of ADRs, with this method, it cannot be the sole method to predict drug effects and a drug may not enter the clinical phase of development by completing only *in vitro* SPP assays: other pharmacological, toxicological, and absorption, distribution, metabolism, elimination assays must be performed as well (5).

As mentioned previously, there are numerous time and cost limitations associated with traditional ADR detection methods. Therefore, computer-aided approaches such as machine learning, data mining, and big data analytics are central to modern ADR detection, all of which exploit electronic health records, clinical trial data, web search logs, and chemical properties of drugs (4). Several algorithms that utilize topological and intrinsic features of drugs to predict novel drug-side effect associations have been proposed and systematically analyzed and compared based on performance (8). Huang *et al.* (9) developed a computational framework to predict drug-ADR associations by combining clinical data with drug-protein and protein-protein interactions. Dey *et al.* (10) successfully introduced a deep learning framework using 2D and 3D representations of drugs to then find chemical substructures that are related to a given ADR and ultimately predict ADR-drug associations. Timilsina *et al.* (11) integrated semantic similarity between drugs obtained through natural language processing (NLP) techniques and known drug-side effect associations to predict unknown ADRs through diffusion of learned weights of side effects. Collectively, these studies have included data to characterize drugs and therefore highlight the influence of drug-protein interactions, protein-protein interactions, drug-drug chemical similarity, and drug-drug semantic similarity on drug-ADR pairs. Together, molecular interaction mechanisms, chemical structural properties, and literature-derived semantic relationships form complex, interacting systems, yet current predictive models and computational frameworks consider them in isolation. Therefore, there is a need for the development of a computational method capable of modeling the multiple factors that contribute to ADRs to ultimately predict unknown drug-ADR associations. Motivated by this, the thesis builds upon the study by Timilsina *et al.* (11) and Yu *et al.* (13) by (i) developing a single heterogeneous graph by integrating known drug-side effect association, drug–drug semantic and chemical structure similarity networks with biological interaction mechanisms, including drug–protein, protein–protein, and drug-drug interactions (ii) constructing and applying a Relational Graph Convolution Network

encoder to allow information passing across to learn features of the nodes of the input heterogeneous graph structure and (iii) reconstructing the drug-side effect network with unobserved, predicted association between drug and side effects.

Before continuing further, it must be specified that the use of terms “ADR” and “side effect” in this paper are used interchangeably. This is due to the term usage of “side effect” by the SIDER database resource that refer to ADRs (7).

Timilsina *et al.* (11) developed a framework to predict links between drugs and side effects. This approach differed from other similar studies due to the use of heterogeneous networks as opposed to homogeneous networks. The method in this study proceeded as follows:

1. Construction of bipartite graph representing drug-side effects associations
2. Creation of drug-drug semantic similarity network using word2vec model trained on PubMed abstracts and PMC full texts
3. Application of Non-negative Matrix Factorization (NMF) to learn diffusion weights of side effects
4. Diffusion of learned weights through heat diffusion model, where drugs linked with side effect information act as a “heat source” that influence neighboring drugs in the similarity network

The combined NMF and heat diffusion framework was found to outperform other state-of-the-art link prediction methods. Key insights from this study are the application and effective integration of the method to a heterogeneous graph, the introduction of predicting associations between entities as a link prediction problem, and that the similarity metric is derived from non-laboratory data.

Yu *et al.* (13) proposed a hybrid GNN model, idse-HE, to discover potential side effects of drugs:

1. Construction of bi-adjacency matrix and bipartite graph of drug-side effect association network
2. Implementation of two GNN models of graph embedding and node embedding to learn chemical information and entity relation information of drugs
3. Make drug-side effect prediction task into a matrix completion task by applying generalized matrix factorization (GMF)

This method was also evaluated and outperformed baseline methods. This approach reinforced the link prediction and matrix factorization method like (11). However, it differed in that a two-GNN method was utilized to aggregate node information to neighboring nodes, and the similarity metric

utilized was that of the drug's chemical structure. Considering the methodology and results of these studies, the aim of this thesis differs in the application of a single R-GCN implementation as opposed to heat diffusion or two-GNN methods. Additionally, this thesis will incorporate more than one relationship type. However, this thesis will frame the drug-side effect association prediction task as a link prediction task, like both studies.

The link prediction task in graph theory is defined as a process of estimating the existence or likelihood of missing links between nodes in a network based on known information such as graph structure and node/edge attributes. Two types of link prediction exist: structural and temporal. This thesis focuses on structural link prediction, in which a partially observed network is provided as input, and the objective is to predict the status of edges for unobserved pairs of nodes (14). To generally characterize this problem, consider an undirected network *G (V, E)* (see Fig. 1) where *V* denotes a set of vertices (nodes) and *E* the set of edges (links). The universal set *U* contains all possible node pairs $\frac{n(n-1)}{2}$ for $n = |V|$. The difference $|U|$ - $|E|$ is representative of the number of non-existing links in the graph (15). However, finding the explicit value of missing links between drugs and side effects from a sparse network of known associations is not the overarching goal; rather, a relevance score is assigned to node pairs that correspond to the probabilistic estimates of unknown associations derived from learned representations of the underlying network. The underlying network must first be represented in a form that exemplifies node relationships.

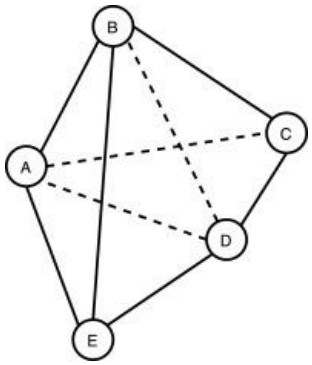

Figure 1. Sample undirected and unweighted network. Link prediction finds missing links indicated by the dotted edges in this sample network (15).

In the context of drug-side effect prediction, the known association network is typically modeled as a bipartite graph, where drugs and side effects are two disjoint node sets connected through observed relationships. A bipartite graph can be generalized as $G = (U, V, E)$ with parts U and V denoting two disjoint sets of vertices (nodes), and edges E between nodes in different sets. Figure 2 shows a sample drug-side effect bipartite graph. In the case of this graph, it must be noted that there are edges (red) between drugs as well, to include the drug-drug similarity. Bipartite graphs

can also be represented as bi-adjacency matrices, in which the elements of the matrix indicate if pairs of nodes are adjacent or not within a graph. Adjacency matrices for bipartite graphs can be defined as follows: for bipartite graph $G = (U, V, E)$ with $|U| = r$ and $|V| = s$, the adjacency matrix is $r \times s$ 0-1 matrix B in which $b_{i,j} = 1$ if and only if $(u_i, v_j) \in E$ (17). Expressing the bipartite drug–side effect network as a bi-adjacency matrix provides a convenient numerical representation that aligns with matrix completion–based approaches to structural link prediction.

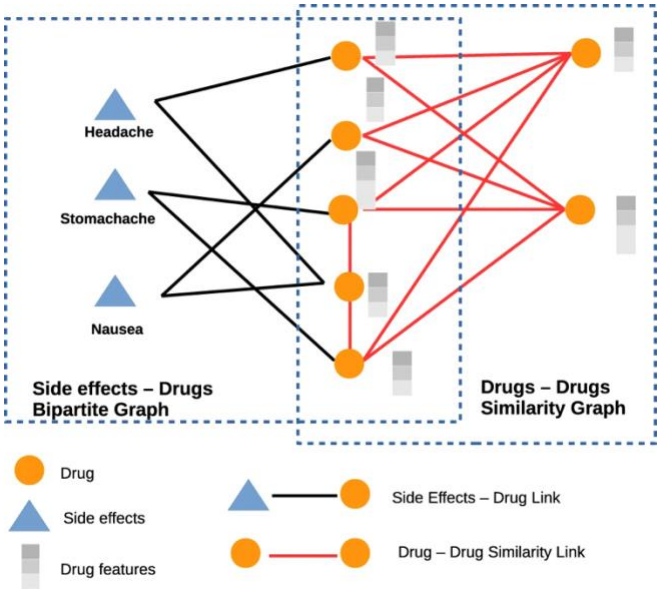

Figure 2. Sample Drug-Side Effect bipartite graph (11).

Building on the bipartite representation of drug–side effect associations, additional biological and chemical relationships can be incorporated by modeling the data as a heterogeneous graph containing multiple node and edge types. A general heterogeneous graph (see Fig. 3) is defined as $G = (V, E, O_V, R_E)$, where V is the set of nodes, E is the set of edges, the type of each node $v \in V$ belongs to the object type set O_V, the type of each edge $e \in E$ belongs to the relation type set R_E (18). The decision to integrate heterogeneous features from heterogeneous data sources to optimize prediction performance is supported by several other studies that accomplish similar problems: Timilsina *et al.* (11) predicts links between side effects and drugs using a heterogeneous graph; Yu *et al.* (18) proposed a drug-target interaction prediction method to learn embeddings of drugs and targets in a heterogeneous network.

Graph Neural Networks (GNNs) are "models that capture the dependence of graphs through message passing between nodes, which take into account the scale, heterogeneity, and deep topological information of input data simultaneously" (19). This deep learning tool has become more prevalent in bioinformatics applications due to the accumulation of biological network data. In

bioinformatics, GNNs can accomplish three levels of graph analysis tasks: node level, edge level, and graph level. Common node level tasks are node classification using semi- supervised learning,

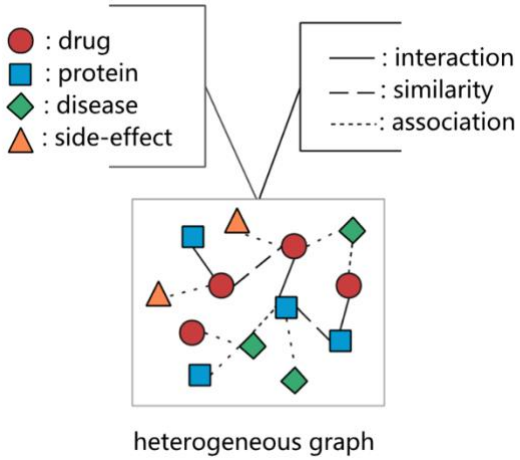

Figure 3. Sample Heterogeneous Graph. Different node and edge types are indicated (18).

an example application being prediction of unlabeled proteins through labeled proteins in protein-protein interaction network. At the graph level, the task at hand is related to graph generation, such as generating synthetic molecules through molecular graph learning. The problem that this thesis addresses falls under the edge level graph analysis: link prediction, as mentioned previously. Several GNN models have been proved to be effective in solving link prediction tasks: Zhang and Chen (20) propose the SEAL (learning from Subgraphs, Embeddings, and Attributes for Link prediction) framework, in which a GNN learns general graph structure features from local subgraphs; Kipf and Welling (21) developed the Graph Convolution Network (GCN) model, a variant of the convolution neural network, which uses layer-wise propagation to encode both graph structure and node features learned on local graph neighborhoods. The Relational Graph Convolution Network (R-GCN) implemented by Schlicktkrull *et al.* (22) can be conceptualized as an extension of the GCN, designed for heterogeneous, multi-relational graphs. Convolution is performed by weighted summation of the output of multiple graph convolutions, and each convolution considers a specific edge type to encode more information from the network structure. Although R-GCNs have typically been applied to directed heterogeneous graphs, the method can still be applied to undirected graphs (such as the heterogeneous graph constructed for the purpose of this thesis) by modeling undirected relations as bidirectional edges. For a heterogeneous graph $G = (V, E, R)$ with node set V, edge set E, and relation types $\mathcal{R}$. The hidden representation of node v_i at layer l is denoted by $h_i^{(l)} \in \mathbb{R}^{d_l}$, where $d_l$is the dimensionality of the layer. For each relation $r \in \mathcal{R}$, $\mathcal{N}_i^r$denotes the set of neighbors of n_i connected under relation r. The R-GCN update rule aggregates messages from relation-specific neighborhoods as:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

where $W_r^{(l)}$is a learnable transformation matrix associated with relation r, $W_0^{(l)}$models self-loop contributions to ensure that the representation of a node at layer $l + 1$ can also be informed by the corresponding representation at layer l , $c_{i,r}$ is a normalization constant and $\sigma(\cdot)$ denotes a nonlinear activation function (22, 23). Although edges in the constructed network may carry additional attributes such as similarity scoring, the formula above describes the general message-passing mechanism, with such attributes incorporated during the methodology stage. To address the issue of performing message passing on highly multi-relational graphs and the fast growth of number of parameters, a weight regularization method known as basic decomposition was defined:

$$W_r = \sum_{b=1}^{B} C_{rb}V_b$$

Where W_r are matrices derived as linear combinations of a smaller set of B basis matrices V_b that are shared across all relations. Therefore, each matrix is a weighted sum of the basis vectors with a component weight C_{rb}; both C_{rb} and V_b can be learned. This method reduces the number of parameters needed to learn the multi-relational data and is a form of effective weight sharing between different edge types (22). Schlicktkrull *et al.* continue to organize the method in an encoder-decoder framework, in which the R-GCN is designated as the encoder to learn node feature embeddings and are utilized by a decoder to produce a score for every edge in the graph. Following this encoder-decoder framework, R-GCN remains the encoder to learn the node feature embeddings from the input heterogeneous graph, which will then be passed to a decoder for reconstructing drug-side effect associations.

To return to the problem of structural link prediction, the learned embeddings produced by the R-GCN encoder are decoded to reconstruct the drug-side effect bi-adjacency matrix, thereby treating the task as a matrix completion problem, as proposed by Menon *et al.* (14) and utilized by (11) and (13). Kumar *et al.* (15) extends this proposal by reviewing several types of matrix factorization link prediction techniques, in which latent unobserved features are extracted and used along with additional node/link information to represent each node in a latent space for link prediction. NMF is the common method in most works, as used by (11), due to its non- negative

interpretable advantage. GMF is another method, as used by (13). Referring to the adjacency matrix B defined earlier, GMF reconstructs the adjacency matrix $B' \in \mathbb{R}^{r \times s}$ defined as follows:

$$B' = E_U L_U (E_V R_V)^T$$

Where $E_U \in \mathbb{R}^{r \times d_{emb}}$ and $E_V \in \mathbb{R}^{s \times d_{emb}}$ denote embedding matrices for nodes in sets U and V and are linearly projected using learnable projection matrices $L_U \in \mathbb{R}^{d_{\text{emb}} \times d_{\text{hid}}}$ and $R_V \in \mathbb{R}^{d_{\text{emb}} \times d_{\text{hid}}}$ (13). GMF projects learned feature matrices of different nodes to low dimensional space for matrix reconstruction with predicted relevance scores between nodes and is an important component of Neural Collaborative Filtering (NCF), which further extends traditional matrix factorization by replacing inner product interactions with neural architecture with the ability of modeling complex relational structures (16). The latter method was therefore chosen as it supports heterogeneous bipartite matrix completion without the non-negativity constraints of NMF, allows drug and side effect embeddings to be projected from graph-derived representations, and utilizes a probabilistic reconstruction of the drug-side effect adjacency matrix.

All code related to the data collection, data integration, data processing, model implementation and visualization can be found in the following GitHub Repository:
https://github.com/clairesprobst/thesis_final

2 Methodology

This thesis presents a heterogeneous graph framework for predicting probability ADRs from a set of known drug-side effect links using R-GCN. The methodology is carried out in four phases (see Fig. 4):

Phase 1: Data Integration

Multiple data sources and computational tool sets were integrated to obtain a more complete set of node and node features, as well as corresponding molecular mechanisms. This includes known drug-side effect association network from drugs and side effects database, drug-protein, protein-protein, and drug-drug interactions from chemical protein interaction databases, drug-drug chemical structure similarity from chemical information database and computational cheminformatics toolkit, and calculated drug-drug semantic similarity from a pre-trained biomedical language representation model.

Phase 1.1: Bipartite and Bi-Adjacency Matrix Construction

The known drug-side effect association network is modeled as a bipartite graph $G = (D, S, E)$ where *D* represents the set of drugs, *S* represents the set of side effects, and *E* denotes the observed associations between them. Edges only exist between nodes belonging to different sets. The bipartite graph *G* is then represented numerically using an adjacency matrix $B = \{b_{ij}\}^{N_d \times N_s}$, where N_d, N_s are the total number of drugs and side effects respectively and $b_{ij} = 1$ if edge (d_i, s_j) exist, otherwise $b_{ij} = 0$. This matrix serves as a foundation for the link prediction task; specifically, zero-valued (unobserved) entries correspond to potential drug-side effect associations to be inferred by the model framework and is used later as a reference for evaluation. Expressing the bipartite network in matrix form enables the application of matrix-based and graph-based learning techniques.

Phase 2: Heterogeneous Graph Construction

The data are aggregated into a heterogeneous graph formally defined as $HG = (V, E, R_E)$, where $V = D \cup S \cup P$ represents set of node types drugs, side effects, and proteins. The edge set *E* is expanded from the original drug-side effect associations to include multiple relation types. The relation type set is $R_E = \{$drug-side effect, drug-protein, drug-drug protein-protein, drug-drug$_{\text{semantic}}$, drug-drug$_{\text{chemical}}\}$. The resulting heterogeneous graph serves as the input to the subsequent learning phase.

Phase 3: R-GCN Construction and Application

To perform link prediction on the constructed heterogeneous graph, an R-GCN as the main learning model, as they support multi-relational graphs by performing relation-specific message passing and therefore are suited for modeling the information present in the integrated input graph. Given the heterogeneous graph $HG = (V, E, R_E)$, the R-GCN encoder updates node representations using the aggregation mechanism defined by Schlichtkrull *et al.* (22) multiplied by the scalar edge weight. At each layer, node embeddings are updated by aggregating transformed messages from neighboring nodes of each relation type. Relation-specific transformation matrices are learned to capture the structural and semantic characteristics of different edge types, including drug–side effect associations, drug–protein interactions, protein–protein interactions, drug–drug semantic similarity, and drug–drug chemical similarity. The resulting drug and side effect embeddings are passed to a decoder.

Phase 4: Bi-Adjacency Matrix Reconstruction

The drug-side effect link prediction task combines the R-GCN encoder with a scoring function decoder. Using the R-GCN output of learned drug and side effect embeddings, a generalized matrix factorization as utilized by Yu *et al.* (13) reconstructs the drug-side effect adjacency matrix such that $B' \in \mathbb{R}^{N^d \times N^s}$ and can be defined as $B' = E_d L_d (E_s R_s)^T$. Matrix multiplication and $(\cdot)^T$ denote standard matrix multiplication and transposition. The drug embeddings and side effect embeddings, $E_d \in \mathbb{R}^{N_d \times d_{\text{emb}}}$ and $E_s \in \mathbb{R}^{N_s \times d_{\text{emb}}}$ respectively, are first linearly projected into a lower-dimensional latent space through learned transformation matrices $L_d \in \mathbb{R}^{d_{\text{emb}} \times d_{\text{hid}}}$ and $R_s \in \mathbb{R}^{d_{\text{emb}} \times d_{\text{hid}}}$ and are then combined elementwise to compute relevance scores for each element $b'_{i,j}$ between drug d_i and side effect s_j pairs. The decoder outputs the continuous-valued reconstruction matrix, where each entry represents the predicted probability of association between drug and side effects.

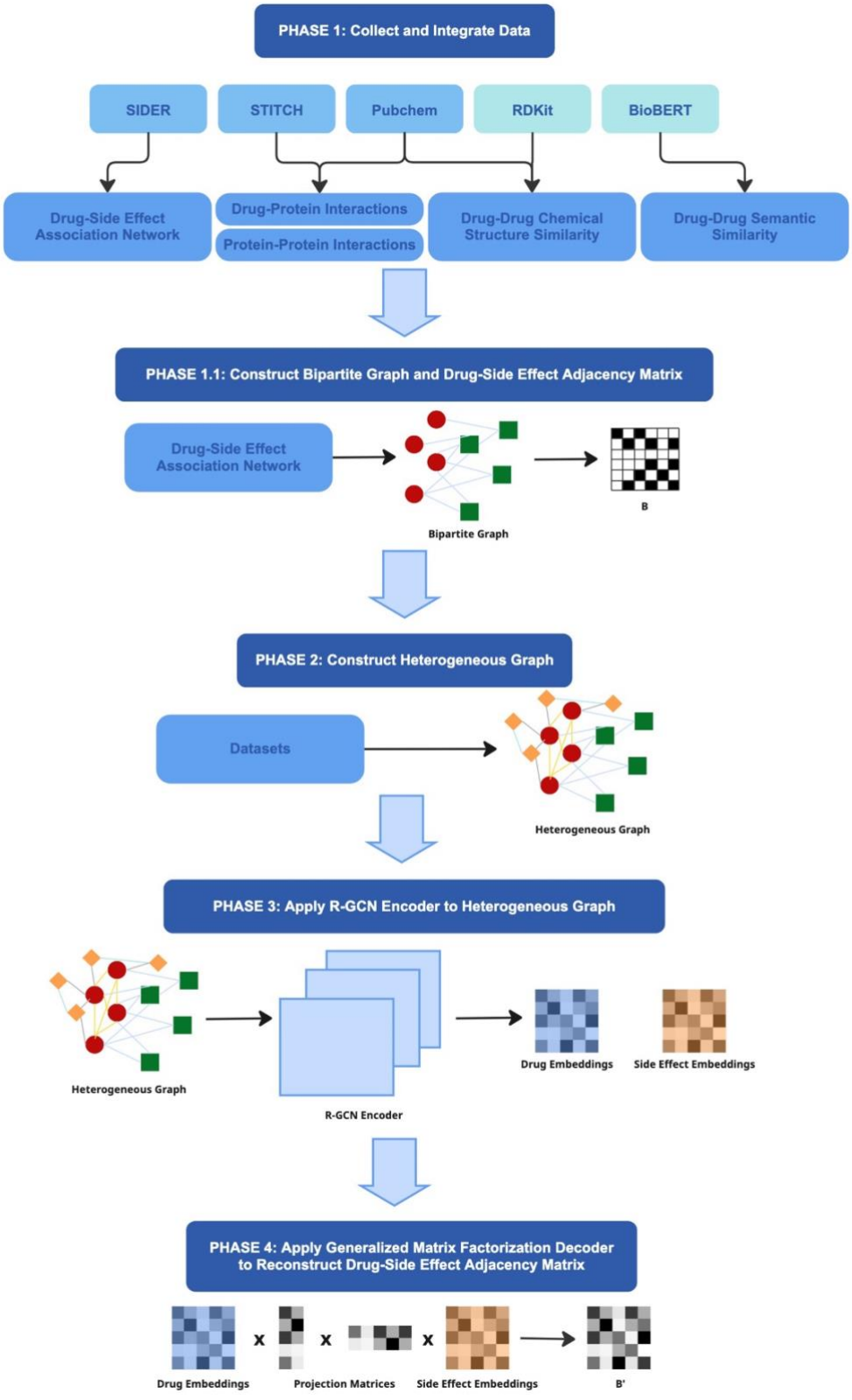

Figure 4. Technical workflow for Heterogeneous Graph Framework for Drug-Side Effect Link Prediction. The pipeline integrates data from public databases and computational tools to drug-side effect bipartite adjacency matrix, which is aggregated with additional datasets into a heterogeneous graph. An R-GCN encoder learns drug and side effect node embeddings from multiple edge types, and generalized matrix factorization decoder reconstructs drug–side effect adjacency matrix to predict previously unknown associations.

3 Data Collection and Dataset Definitions

The objective of the following section is to collect, transform, and preprocess data from several publicly available chemical and biomedical data sources systematically to construct datasets describing drug properties and their associated molecular mechanisms for heterogeneous network construction. The datasets define the node entities of the network: drugs, side effects, and proteins, and the relationships between them, such as drug-side effect associations, drug-drug similarity, drug-protein, drug-drug, and protein-protein interactions. In addition to defining the network structure, relevant biological, chemical, and semantic attributes associated with each node entity were obtained to provide descriptive feature information. The collected node entity attributes were transformed to numerical feature representations using embedding methods to enable their integration into the R-GCN framework. This section provides detailed descriptions of the data sources, entity and relationship definitions, attribute collection procedures, and feature representation methods used to construct the final datasets. These datasets collectively form the structural and feature-level foundation of the heterogeneous graph used for subsequent network construction and graph neural network modeling.

3.1 Overview of Data Sources and Computational Tools

The purpose of this section is to define the data sources, databases, and computational tools utilized to obtain the node entities, their attributes, and the relationships between the nodes.

3.1.1 Datasets

BioSNAP (24) is a part of the Stanford Network Analysis Project (SNAP) and contains a collection of large-scale biomedical network datasets for applications in network biology, bioinformatics, and machine learning. BioSNAP includes networks representing relationships among biological and biomedical entities including drugs, side effects, proteins, genes, cells, genomic regions, diseases and functions. This thesis utilizes the drug-side effect association network to define links between drug and side effect nodes. https://snap.stanford.edu/biodata/

The Side Effect Resource (SIDER) 4.1 database (7) is a repository that aggregates information on adverse drug reactions extracted from public documents, package inserts, drug labels, and adverse reporting systems that collect reports from doctors, patients, and drug companies. SIDER provides a standardized mapping between drugs and their associated side effects using the STITCH identifiers

(drugs) and Unified Medical Language System (UMLS) Concept Unique Identifiers (CUI). The drug-side effect associations provided in the BioSNAP dataset originate from the SIDER database. http://sideeffects.embl.de/

The Search Tool for Interactions of Chemicals (STITCH) database (12) is a resource for the known and predicted interactions of chemicals and proteins. The interactions include both physical and functional associations and are derived from genomic context prediction, high-throughput lab experimentation, co-expression, text mining from of bibliographic and gene databases (MEDLINE and Online Mendelian Inheritance in Man, respectively), and previous knowledge in databases. When querying the STITCH database using STITCH identifiers, a visual network of related proteins and chemicals to the queried identifier is presented containing edge types for binding, activation, and inhibition are presented, along with a likelihood of interaction score between each pair of entities in the network (see Figure 5). https://stitch-db.org/

PubChem (25) is an open chemistry database maintained by the National Institutes of Health of compounds, substances, and bioassays. Each entry has comprehensive information including molecular structure, chemical identifiers, physiochemical properties, and biological annotations. PubChem integrates data from multiple authoritative sources and assigns unique Compound Identifiers (CID) to chemical entities, enabling consistent referencing across chemical and biomedical datasets. The database serves as a widely used resource for chemical informatics, drug discovery, and computational biomedical research. https://pubchem.ncbi.nlm.nih.gov/

The Universal Protein Knowledge Base (UniProtKB) is a data resource for protein sequence and their annotations (26). Experimentally validated and computationally predicted data, including amino acid sequences, protein names, gene associations, functional descriptions, and cross-referencing with other databases are integrated and together serve as a resource for protein-related information. https://www.uniprot.org/

3.1.2 Computational Tools

BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) is a pre-trained language representation model for biomedical text mining based on the Bidirectional Encoder Representations from Transformers (BERT) architecture (27). BioBERT extends the general-purpose BERT model by pre-training on biomedical domain corpora (PubMed abstracts and PMC full-text articles). BioBERT therefore captures semantic and contextual relationship between

biomedical terms more effectively than the general domain language model. BioBERT has been used in biomedical natural language processing (NLP) tasks including entity recognition, relation extraction, and semantic similarity analysis. This thesis utilizes BioBERT to generate semantic feature representations between drug names and thus incorporating semantic similarity information into the heterogeneous network framework. https://github.com/dmis-lab/biobert-pytorch

RDKit is an open-source cheminformatics software that provides functionality to read, represent, process, and analyze chemical structures (28). Cheminformatics include but are limited to molecular structure parsing, molecular fingerprint generation, similarity computation, and enables the

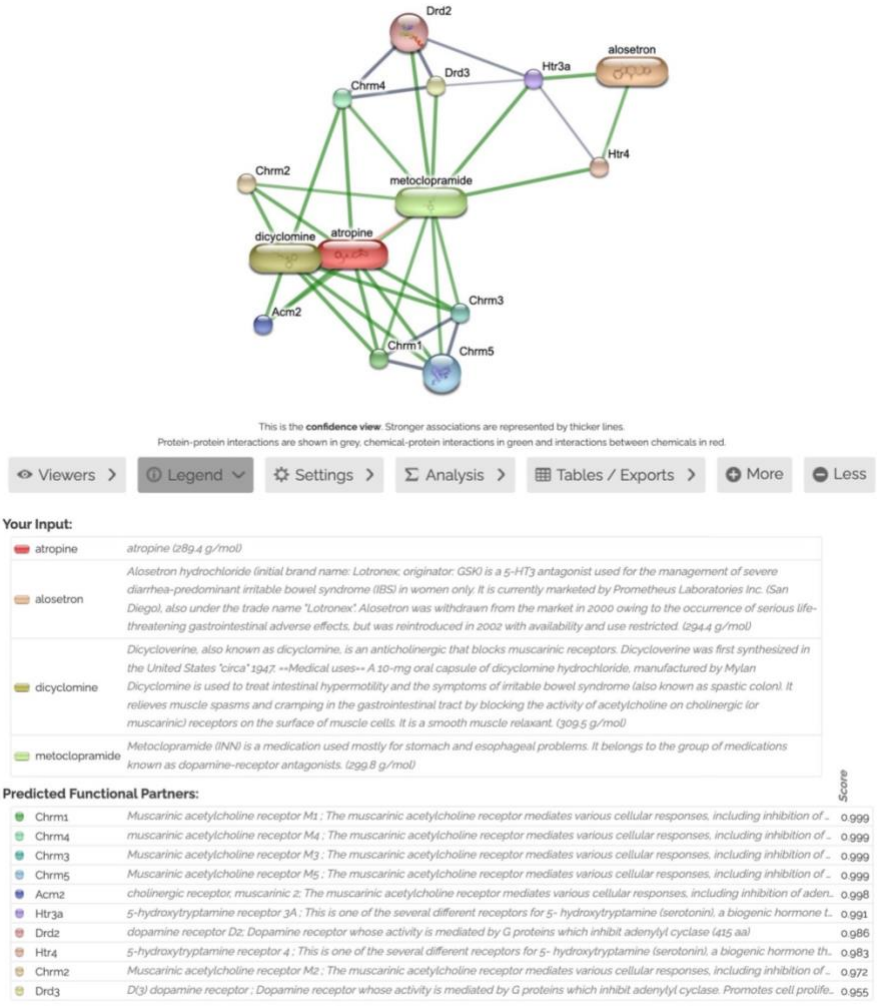

Figure 5. Interaction Network of atropine, aldosterone, dicyclomine, and metoclopramide from drug-side effect dataset from STITCH. Drugs are represented as pill-shaped nodes, proteins shown as spheres. Network edges indicate an interaction may be present, and the thickness of the edge represent the confidence score, i.e. how likely an interaction is given to be true. Edge types in green represent activation, red represents inhibition, and blue represents binding, and thickness of edge. The list of proteins ('Functional Partners') are listed and described. STITCH can query individual entries or multiple entries, as shown (12).

conversion of chemical structure representations, such as Simplified Molecular Input Line Entry System (SMILES) strings, into machine-readable molecular objects and numerical feature representations. As it is easily implemented in Python and has extensive functionalities, it is a tool commonly used for chemical similarity analysis, as performed in this thesis. https://www.rdkit.org/

The pre-trained clinical concept embedding framework cui2vec represents medical concepts as dense numerical vectors based on co-occurrence patterns in biomedical data (29). The model maps medical concepts to a shared embedding space using CUIs from the UML. Such embeddings were learned from biomedical corpora of clinical notes, insurance claims, and biomedical literature and allows for semantically and clinically related concepts to be positioned closer in vector space. cui2vec

ESM-2 8M is a protein language model for amino acid sequence-level analysis, designed to generate numerical representations of protein sequences (30). Similar to BERT-style models, it is a bi-directional encoder trained on large-scale protein sequence databases to learn contextual relationships between amino acids. ESM-2 can obtain structural and functional properties of proteins directly from their amino acid sequences. In this thesis, ESM-2 was used to generate the embedding representations of protein amino acid sequences, used as feature attributes for protein nodes in the heterogeneous network. esm

3.2 Node Entity and Relationship Data Collection

The purpose of this section is to describe how the node and edge sets that define the heterogeneous

network are constructed. Using the biomedical databases described in Section 3.1, entities corresponding to drugs, side effects, and proteins were identified and used to define the node sets of the network. Relationships between these entities, including drug–side effect associations, drug–protein interactions, protein–protein interactions, and drug–drug similarity relationships, were collected to define the edge sets. In addition to edge connectivity, several relation types include associated edge weights. Formally, each relation-specific edge set may be associated with a weight function: $w_r: E_r \rightarrow \mathbb{R}$ which assigns a scalar weight to each edge based on the corresponding interaction confidence or similarity measure. Together, the sets form the foundation of the heterogeneous graph used for R-GCN modeling.

3.2.1 Drug Entity Collection

The drug entity set was constructed using the drug-side effect association dataset from BioSNAP (24), specifically the *ChSe-Decagon_monopharmacy.csv* file. The dataset was derived from the STITCH database and contains associations between a set of drugs using the STITCH compound identifier and their reported side effects. Formally, the drug entity set is defined as $D = \{d_1, d_2, \ldots, d_{|D|}\}$ where each drug $d_i \in D$ is uniquely identified by its STITCH identifier. This drug entity set contains $|D|$=639 and forms one component of the heterogeneous graph G defined in Section 2. The descriptive information of each drug entity, namely drug names, were retrieved through an automatic data retrieval script the SIDER and PubChem databases using `beautifulsoup4` and `Selenium`. The resulting drug entity set and associated identifiers were used to define the drug nodes in the heterogeneous network and can be found in *nodes_drugs.csv.*

3.2.2 Side Effect Entity Collection

The side effect entity set was also constructed with the *ChSe-Decagon_monopharmacy.csv* file. Formally, the side effect entity set is defined as $S = \{s_1, s_2, \ldots, s_{|S|}\}$ where each side effect $s_i \in S$ is uniquely identified by its UMLS Concept Unique Identifier. This side effect entity set forms another component of the heterogeneous graph *HG* defined in Section 2. Side effect names corresponding to each CUI were obtained directly from the BioSNAP dataset, which was originally derived from the SIDER database, which contained $|S|$ = 10184 unique side effects. These identifiers and associated names were used to define the side effect nodes in the heterogeneous network and can be found in *nodes_se.csv.*

3.2.3 Protein Entity Collection

The protein entity set was constructed was constructed using interaction data obtained from the STITCH database (12). To retrieve protein interaction data relevant to the drug entity set *D,* drugs were grouped according to their Anatomical Therapeutic Chemical (ATC) classification codes at the second level of the ATC hierarchy. The drug groupings were used to perform batch querying to STITCH to efficiently retrieve drug-protein, protein-protein, and drug-drug interaction data. For drugs without ATC classification codes, individual queries were performed to ensure coverage of the drug entity set. The results contained interaction pairs between drugs and proteins as well as between proteins, along with their corresponding identifiers and interaction confidence scores. Protein entities

were identified using their ENSEMBL protein identifiers. To ensure high confidence in the interaction data and reduce noise, only interactions with a combined confidence score greater than or equal to 0.7 were retained. Formally, the protein entity set is defined as $P = \{p_1, p_2, \ldots, p_{|P|}\}$ where each protein $p_i \in P$ is uniquely identified by its ENSEMBL protein identifier. The final protein entity set after filtering consists of $|P| = 502$ unique protein nodes. This protein entity set forms the final component of the heterogeneous graph G defined in Section 2. These protein entities serve as the basis for constructing drug–protein, protein–protein, and drug-drug relationships in the heterogeneous network and can be found in *nodes_protein.csv.*

3.2.4 Drug-Side Effect Relationship Collection

Drug–side effect relationships were obtained from the BioSNAP dataset file *ChSe-decagon_monopharmacy.csv*. Each record represents an observed association between a drug and a side effect. Formally, this relation corresponds to the relation types $r_{\text{drug–side effect}} \in R_E$, with the associated edge to the edge set defined as $E_{\text{drug–se}} \subseteq \mathrm{D} \times \mathrm{S}$ where each edge $(d_i, s_j) \in E_{\text{drug–se}}$represents an association between drug $d_i$and side effect s_j. Edges between drugs and side effects are unweighted. This relation-specific edge set forms one component of the overall edge set $E$of the heterogeneous graph $HG = (V, E, R_E)$ and can be found in *edges_drug_se.csv.*

3.2.5 Drug-Drug Semantic Similarity Relation Collection

Drug–drug semantic similarity relationships were computed using semantic embedding representations generated from biomedical text using the BioBERT pretrained language model. Semantic similarity between drug pairs was calculated using cosine similarity. Formally, this relation corresponds to the relation type $r_{\text{drug-drug semantic}} \in R_E$, with the associated edge set defined as $E_{\text{semantic}} \subseteq \mathrm{D} \times \mathrm{D}$ where each edge $(d_i, d_j) \in E_{\text{semantic}}$represents a semantic similarity relationship between drugs $d_i$and d_j. Each edge $(d_i, d_j)$is associated with a cosine semantic similarity score computed from BioBERT embeddings, which was used as the edge weight. The edge set was filtered to retain only edges with similarity scores greater than or equal to 0.88. This relation-specific edge set forms one component of the overall edge set E and can be found in *edges_drug_drug_semantic.csv.*

3.2.6 Drug-Drug Chemical Similarity Relation Collection

Drug–drug chemical similarity relationships were computed using Simplified Molecular Input Line Entry System (SMILES) representations (see Fig. 6) obtained from PubChem to processed using RDKit. SMILES are a form of line notation to describe drugs using ASCII strings (31), and with RDKit each SMILES string of drug nodes is converted to a molecular object, and an extended connectivity fingerprint (ECFP) with radius 2 and length 1024 bits were generated to represent the

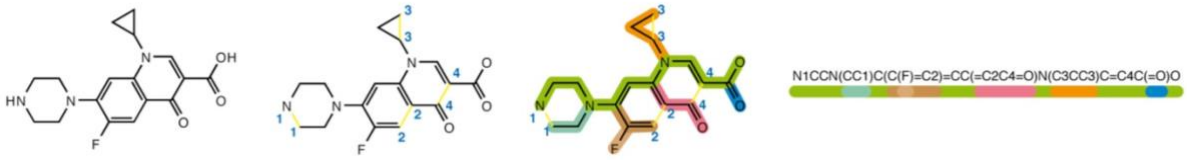

Figure 6. SMILES representation of ciprofloxacin. 2D representation on left to SMILES representation on right. Algorithm break cycles, then writes as branches off a main backbone (31). Color is for clarity.

chemical structure of each drug. ECFPs (referred to as Morgan fingerprints in code) essentially turn a molecule into an informative vector representation (32), and therefore this fingerprint is used as input to generate the pairwise chemical similarity score between drug pairs using Tanimoto similarity coefficient. Formally, this relation corresponds to the relation type $r_{\text{drug-drug chemical}} \in R_E$, with the associated edge set defined as $E_{\text{chemical}} \subseteq \text{D} \times \text{D}$ where each edge $(d_i, d_j)$is associated with a Tanimoto similarity score representing the chemical similarity between drugs $d_i$and d_j, which was used as the edge weight. The edge set was filtered to retain only edges with similarity scores greater than or equal to 0.30. This relation-specific edge set forms one component of the overall edge set E and can be found in *edges_drug_drug_chemical.csv.*

3.2.7 Drug-Protein, Protein-Protein, and Drug-Drug Relation Collection

Drug–protein, protein-protein, and drug-drug relationships were obtained from the STITCH database as described in Section 3.2.3. The resulting interaction data are filtered to retain only interactions with a combined confidence score greater than or equal to 0.70. Formally, these relations correspond to the relation types $r_{\text{drug–protein}}$, $r_{\text{protein–protein}}$, $r_{\text{drug–drug}} \in R_E$, with the associated edge set defined as: $E_{\text{dpi}} \subseteq \text{D} \times \text{P}$, ; $E_{\text{ppi}} \subseteq \text{P} \times \text{P}$; $E_{\text{ddi}} \subseteq \text{D} \times \text{D}$ where each edge $(d_i, p_k) \in E_{\text{dpi}}$ represents an interaction between drug $d_i$and protein p_k, each edge $(p_l, p_k) \in E_{\text{ppi}}$ represents an interaction between protein $p_i$and protein p_k, and each edge $(d_i, d_j) \in E_{\text{ddi}}$ represents an interaction between drug $d_i$and drug d_j. Each edge is associated with a STITCH combined confidence score, which was used as the edge

weight. These relation-specific edge sets form three components of the overall edge set E of the heterogeneous graph HG and can be found in *edge_drug_protein_mixed.csv*.

3.3 Node Attribute Collection

The purpose of this section is to describe how the attributes of each node entity were obtained.

3.3.1 Drug Attributes

Drug attribute information, namely Anatomical Therapeutic Chemical (ATC) classification codes, was obtained from the PubChem database using the drug's STITCH compound identifier. ATC codes provide a hierarchical classification system that categorizes drugs according to their therapeutic use and pharmacological properties into five group classification levels. For each

A	Alimentary tract and metabolism (1st level, anatomical main group)
A10	Drugs used in diabetes (2nd level, therapeutic subgroup)
A10B	Blood glucose lowering drugs, excl. insulins (3rd level, pharmacological subgroup)
A10BA	Biguanides (4th level, chemical subgroup)
A10BA02	metformin (5th level, chemical substance)

Figure 7. Complete ATC classification of metformin (33).

Drug $d_i \in D$, ATC classification is organized by full ATC code and corresponding levels 1, 2, 3, and 4 categories (see Fig. 7). From level 1 to full ATC code, the therapeutic classification becomes increasingly specific. However, not all drugs in the dataset returned ATC codes, but were retained to maintain the completeness of the drug-side effect association network. The collected ATC classification data is stored as categorical attributes for each drug in *nodes_drug.csv***.**

3.3.2 Side Effect Attributes

The side effect unique identifier CUI outlined in Section 3.2.2 is also utilized as the main attribute for each side effect $s_i \in S$. They are later used to obtain semantic feature representations using the cui2vec tool, as described in Section 3.4.2.

3.3.3 Protein Attributes

Protein attribute information was obtained from the UniProt database using ENSEMBL protein identifiers corresponding to each protein entity $p_k \in P$. For each protein, the amino acid sequence and sequence length were retrieved to structural information about the protein. In addition, Gene Ontology (GO) annotations were obtained, including cellular component, molecular function, and biological process categories, which describe the protein's location, activity, and functional role within biological systems. These attributes were stored for each protein node and used to characterize protein entities in the heterogeneous graph in *nodes_protein.csv.*

3.4 Node Attribute Embedding and Feature Representation

To utilize the node attributes in the R-GCN framework, raw attributes were transformed to numerical feature representations using pre-trained embedding models and learned embedding layers. The purpose of this section is to describe how the fixed-length vector representations were encoded for each node type to be integrated in the heterogeneous graph.

3.4.1 Drug Feature Representation

Drug node features were constructed using the ATC classification codes collected in Section 3.3.1. Because ATC codes are categorical attributes across hierarchical levels, each ATC code level was converted into integer labels through sklearn `LabelEncoder()`. The encoded attributes were then transformed to dense vector representations using learned embedding layers where `emb_dim`=16, implemented with PyTorch. This method was used as opposed to one-hot encoding, which would produce sparse, high-dimensional vectors that treats all categories equally distant. Separate embedding layers were used for each of the four ATC levels and allow the model to learn compact representations that capture latent similarities between drug categories, and each level constitutes a distinct categorical vocabulary with its own cardinality. The embeddings were eventually concatenated into a single feature vector for each drug node. The embedding layers were used to create fixed feature representations and used as input features for drug nodes of *HG.*

3.4.2 Side Effect Feature Representation

Side effect node features were constructed using both pre-trained cui2vec and BioBERT model embeddings. Combining both enables the integration of clinical concept relations from cui2vec and

the semantic embeddings between side effect names from BioBERT to give side effect nodes a richer set of features. The cui2vec embeddings were used to obtain semantic vector representations based on each unique identifier. The resulting embeddings were concatenated to form a combined feature vector for each side effect node of size [10184, 1268]. However, before training the model, the vectors were reduced to a 64-dimensional representation to reduce computational cost of training the R-GCN. These were used as input features for side effect nodes of *HG*.

3.4.3 Protein Feature Representation

Protein node features were constructed by combining sequence based and annotation-based representations. The amino acid sequences obtained from UniProt were embedded using the pre-trained ESM-2 protein language model (facebook/esm2_t6_8M_UR50D). The model tokenized each protein sequence and aggregated to a fixed-length sequence embedding using mean pooling. To make this process more efficient, the sequence embeddings were generated in small batches and saved in chunks. The protein sequence length was also considered as a numerical attribute and therefore standardized. Lastly, the gene ontology annotations were treated as categorical features: each field was label-encoded with sklearn `LabelEncoder()` and mapped to a learned embedding vector `emb_dim`=8, like that of Section 3.4.3. The final protein feature vector for each node contained the ESM-2 sequence embedding, normalized length attribute, and the GO embedding representations by concatenation to produce a feature vector of size [502, 345] and used as input features for protein nodes of *HG*.

3.5 Dataset Summary

Tables 3.1—3.3 summarize the node types, relation types, and node attributes comprising the datasets used in this thesis.

Node Type	*Symbol*	*Count*	*Identifier*	*Source*	*Attribute Source*
Drug	D	$N_D = 639$	STITCH CID	BioSNAP / SIDER	PubChem
Side Effect	S	$N_S = 10184$	UMLS CUI	BioSNAP / SIDER	BioSNAP / SIDER
Protein	P	$N_P = 502$	ENSEMBL	STITCH	UniProt

Table 3.1 Summary of Node Types

Relation Type	*Symbol*	*Count*	*Source*	*Edge Weight*
Drug-Side Effect	$E_{drug-se}$	174977	BioSNAP / SIDER	None
Drug-Drug Semantic	$E_{semantic}$	18522	BioBERT	Cosine Similarity
Drug-Drug Chemical	$E_{chemical}$	2074	PubChem / RDKit	Tanimoto Similarity
Drug-Protein Interaction	E_{dpi}	1601	STITCH	Confidence Score
Protein-Protein Interaction	E_{ppi}	1082	STITCH	Confidence Score
Drug-Drug Interaction	E_{ddi}	3240	STITCH	Confidence Score

Table 3.2 Summary of relation types

Node Type	*Attribute Components*	*Final Dimension*
Drug	ATC embeddings	64
Side Effect	BioBERT, cui2vec embeddings (reduced)	64
Protein	ESM-2 sequence embedding, length, GO embeddings	345

Table 3.3 Summary node feature representations

3.6 Exploratory Data Analysis of Datasets

An exploratory data analysis was performed on the Drug-Side Effect network, Drug-Drug semantic smilarity network, Drug-Drug Chemical Structure similarity network, and Drug-Protein/Protein-Protein/Drug-Drug interaction network using Python and Gephi application. `Networkx`, `matplotlib`, `seaborn`, `rdkit`, `sknetwork`, and `umap` are the primary python libraries imported to complete the analysis. The full analysis with annotations and more visualizations can be found in *EDA.ipynb.*

3.6.1 Drug-Side Effect Network

The drug-side effect network was constructed to a NetworkX `Graph()` object. Table 3.4 contains graph statistics most notably the bipartite structure and average degree per node. The density of the graph being around 0.003 indicates a sparse network. Observed drug-side effects account for 2.689% of the entire sample set, considering there are 639 drug nodes and 10184 side effect nodes.

A visualization of the bipartite network as a bi-adjacency matrix can be seen in Figure 7. Each filled in cell indicates a link between a drug and side effect, and the drugs were ordered from lowest degree to highest degree, resulting in the stark curve through the matrix visualization. Darker bands can be

observed on both the x-axis and y-axis of the matrix: along the x-axis indicates side effects with higher degree, therefore are associated with higher number of drugs; along the y-axis indicates drugs with higher degree, therefore are associated with higher number of side effects. From this visualization alone the specific drug or side effect is associate with a band can determined, but degree distribution and returning a tabular format of nodes and degrees is helpful.

Attribute	Value
Total Nodes	10823
Total Edges	174977
Density	0.002988
Is Bipartite	True
Drug Nodes	639
Side Effect Nodes	10184
Average Degree	32.334288

Table 3.4 Drug-Side Effect Network Graph Statistics

After plotting (see Figure 8), the degree distribution of drug nodes is left skewed with a long tail to the right, indicating majority of nodes have a lower degree value (side effect link) with few drugs having >900 side effects.

The most reported side effects are returned by sorting side effect node degree values from highest to lowest and are expectedly more general side effects, including examples such as ‘general physical health deterioration’ (associated with 301 drugs), ‘mental status changes’ (associated with 278 drugs) and ‘condition aggravated’ (associated with 270).

Returning to the bipartite structure of the drug-side effect matrix and bi-adjacency matrix representation, community detection was performed using sknetwork’s `from_edge_list()` function and `Louvain()` clustering algorithm. The purpose of this analysis was to identify communities in the bipartite network based on modularity optimization; by assigning each node to be in its own community and for each node it tries to find maximum positive modularity gain by moving each node to all its neighbor communities (35, 36). The `Louvain()` algorithm identified six communities containing both drugs and side effects, and the resulting graph with community information was saved and loaded to Gephi for visualization. Though the full bipartite graph is trivial to analyze, analyzing by community 0-5 individually was much more informative. Figure 9 is one of six analyses completed, where a visual of the community network is visible, along with the most

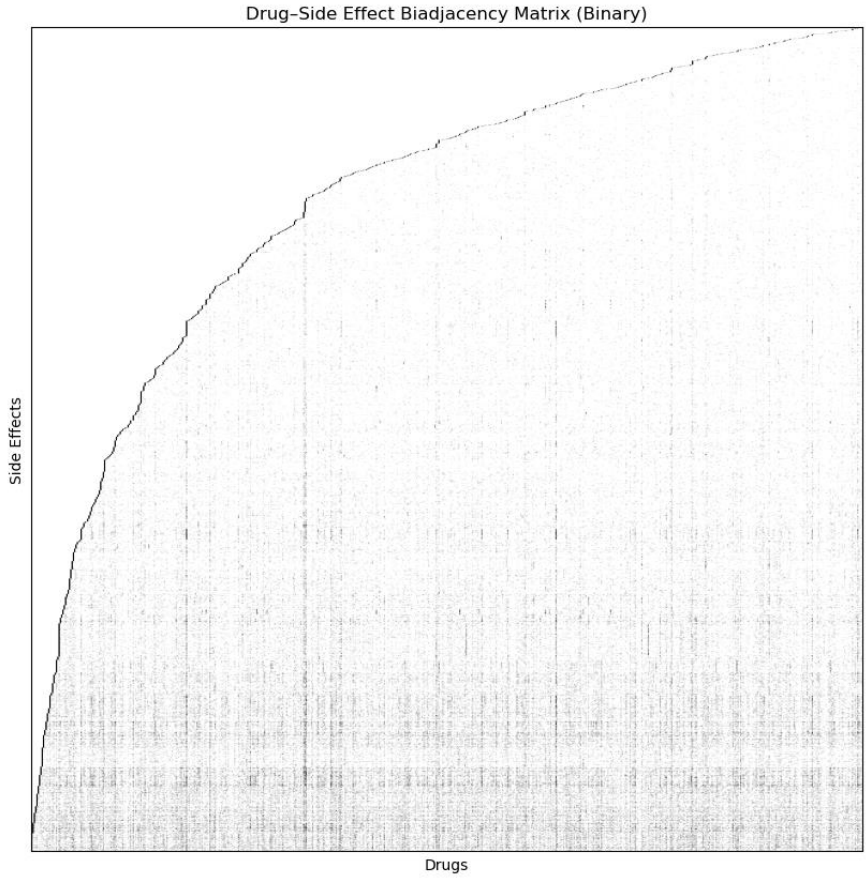

Figure 7. Complete Drug-Side Effect Bi-Adjacency Matrix B. Cell filled if link between drug and side effect, empty if none

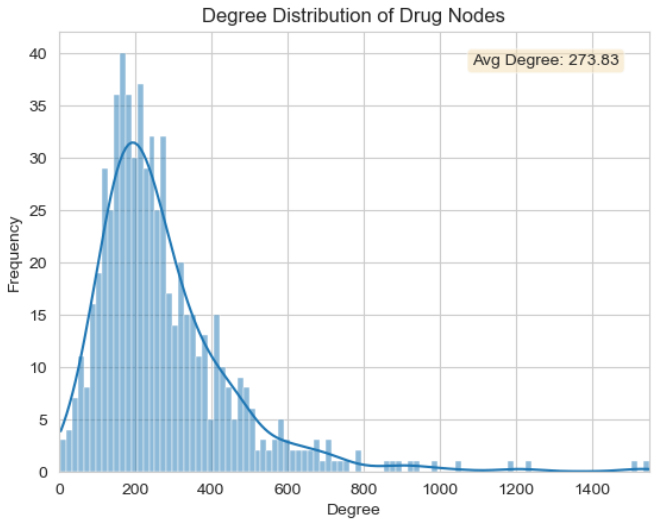

Figure 8. Degree Distribution of Drug Nodes from Drug-Side Effect Network.

common side effects in the community, degree distribution of drugs in the community, and the ranked ATC level 2 frequency. In the case of Community 0, the community is tightly clustered in the visual, and the top two ATC level two classes of drug classification belong to the N06 psych analeptics therapeutic subgroup and N05 psycholeptics therapeutic subgroup containing antidepressants, anti-dementia drugs, and anti-psychotics.

Figure 9. Community Detection using Louvain clustering algorithm on drug-side effect network.

Communities 1-5 also returned a visual like that of Figure 9, which can be found in the referenced jupyter notebook.

As mentioned in Section 3.3.1, the ATC codes contain five classification levels, in which level 1 and level 2 are the most general. Therefore, an analysis of the frequency of ATC level 1 and ATC level 2 classifications within the drug nodes was performed. The three most common ATC level 1 class drugs in the dataset belong to class N (nervous system), class C (cardiovascular system) and type J (anti-infectives for systemic use). ATC level 2 drug classification is more refined, and the frequency graph and statistics regarding ATC level 2 classification of drugs in the dataset can be observed in Figure 10.

3.6.2 Drug-Drug Semantic Similarity Network

The drug-drug semantic similarity is a fully connected graph with density = 1. The analysis of the network concludes with determining the optimal score threshold to retain highly similar drug pairs.

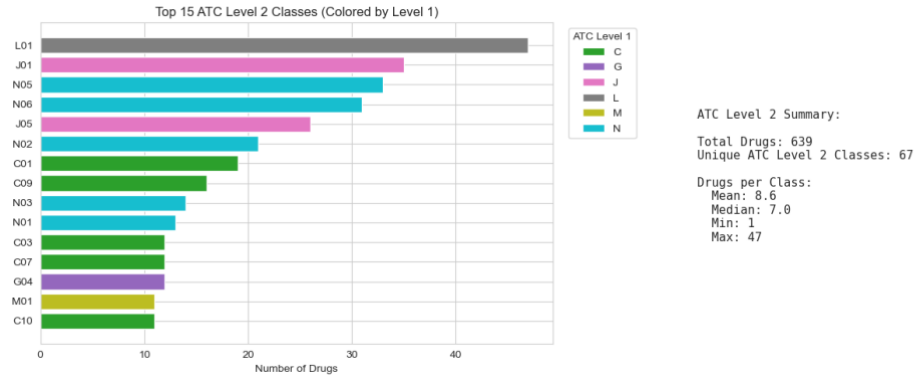

Figure 10. ATC level 2 classification frequency graph and statistics.

The cosine semantic similarity score distribution can be observed in Figure 11. There is a slightly longer left tail and a strong peak above 0.8, indicating that many drug pairs have high similarity (>0.75). However, a threshold value must be determined to reduce the number of edges in the network, that will then be utilized to construct *HG.* To achieve this, several threshold values were used as input to `graph_stats_at_threshold()` function that filtered at the threshold and returned the number of nodes, number of edges, number of components, and average degree at each threshold value analyzed. For the drug-drug semantic similarity network, the threshold is determined to be 0.88, in which 4.54% of edges were retained. Table 3.6 contains the filtered graph network statistics.

3.6.3 Drug-Drug Chemical Structure Similarity Network

The drug-drug chemical similarity network is a fully connected graph with density = 1. The analysis of the network concludes with determining the optimal score threshold to retain highly similar drug pairs.

Prior to filtering, the Butina clustering method is implementing using RDKit's Butina.ClusterData that exploits the calculated Tanimoto value assigned between drug pairs, as described in Section 3.2.6, using a distance matrix and distance threshold to form clusters. To achieve this from the input .csv file of drug chemical similarity, the bit string representation of Morgan fingerprints for each drug was converted to a valid RDKit `DataStructs` fingerprint object to be identified as a molecule. The function `butina_cluster()` was performed on drugs pairs that had a Tanimoto similarity

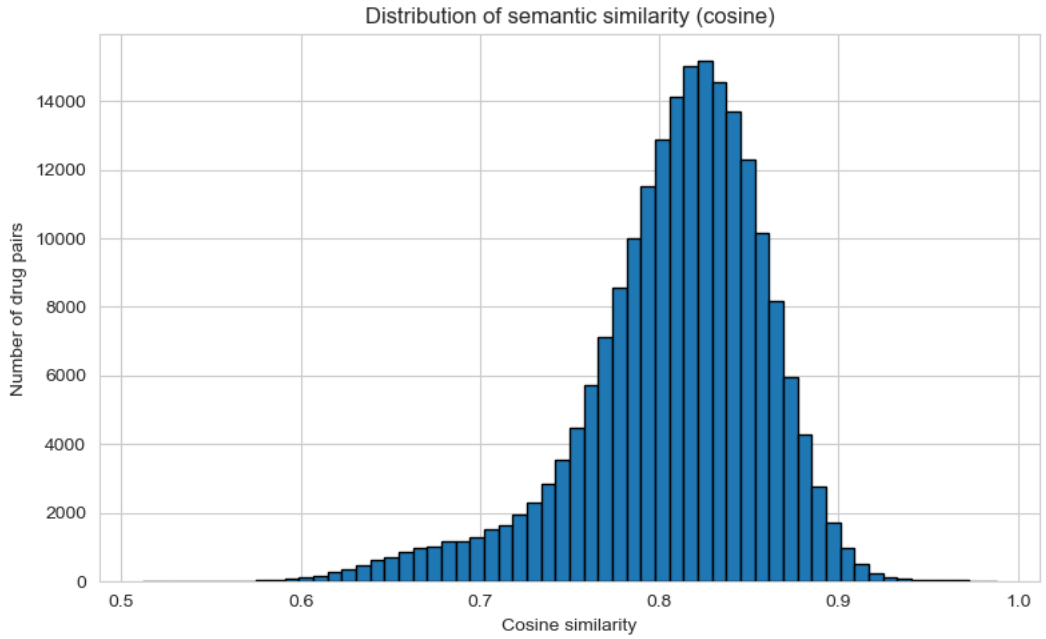

Figure 11. Visualization of similarity score distribution. For drug-drug semantic similarity network.

score of at least 0.5. On 639 drugs, the clustering algorithm returned 465 clusters; the largest clusters containing nine drugs. 78.9% of clusters were singletons (e.g. contained only one drug). A UMAP visualization of the clusters can be observed in Figure 12, in which high dimensional drug fingerprint representations were reduced to 2D coordinate space to visualize clusters (38). The largest identified clusters are highlighted, while the singletons are shaded in gray.

An interesting analysis of a single cluster can be performed by using RDKit to draw the chemical structures of each drug in each cluster. Focusing on the largest identified cluster, there are clear structural similarities in the chemical structures of the drugs within (see Figure 13). They all contain a beta-lactam ring (39) and belong to the ATC level 3 hierarchy J01D: antibacterials for systemic use and other beta lactam antibacterials. More specifically, all the drugs also contain a 6-member dihydrothiazine ring, typically fused with the beta-lactam ring (40). This is visual conformation that The Butina cluster method correctly clustered drugs based on similarity score.

As performed in the previous section, a threshold value must be determined to reduce the number of edges in the chemical structure similarity network, that will then be utilized to construct *HG.* Again, `graph_stats_at_threshold()` function was used. The optimal threshold was determined to be 0.30, and the filtered graph summary statistics can be found in Table 3.5.

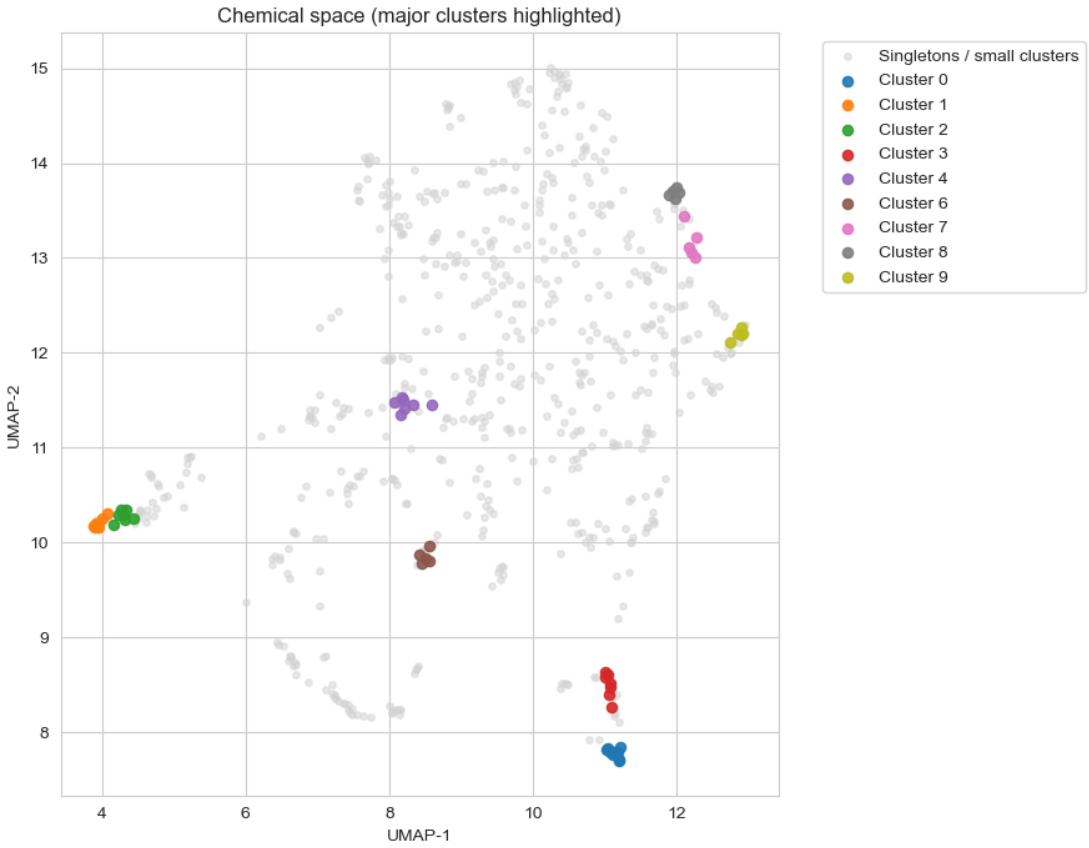

Figure 12. UMAP Plot of Bustina Clustering Results. Performed on drug-drug similarity network.

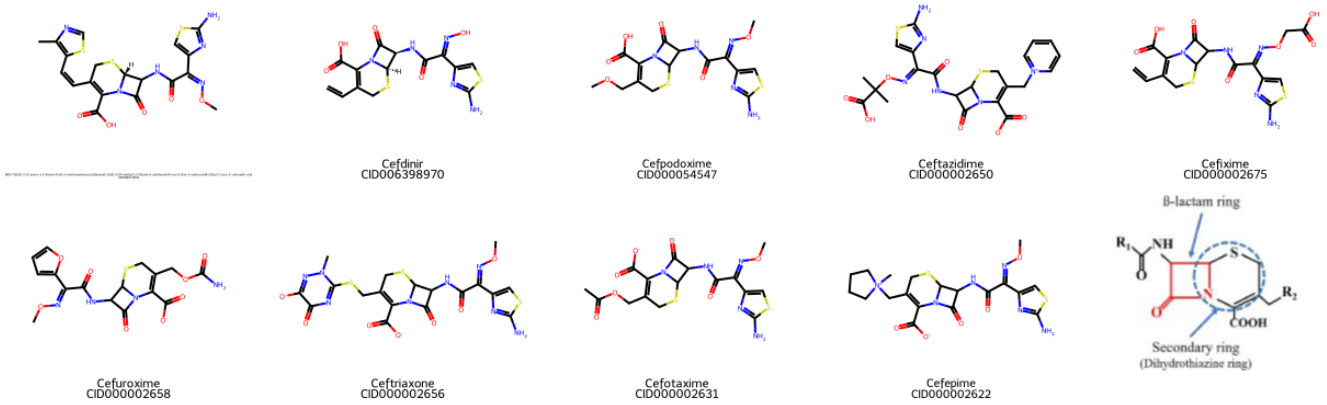

Figure 13. Drawings of chemicals in largest cluster from Butina Clustering algorithm. Each drug contains a beta-lactam ring fused with a dihydrothiazine ring, an example on the second row on the far right.

Attribute	Semantic Similarity Filtered	Chem. Struct. Similarity Filtered
Threshold	0.88	0.30
Total Nodes	604	491
Total Edges	9261	1037
Density	0.050854	0.008620
Average Degree	30.66556	4.22403

Table 3.5. Filtered Semantic Similarity and Chemical Structure Similarity Network Statistics.

3.6.4 Drug-Protein, Protein-Protein, Drug-Drug Interaction Network

The interaction network was made to a graph object `Graph()`. The interaction network contains three different relation types: 2805 drug-drug, 1930 drug-protein, and 977 protein-protein before filtering. From the degree distribution graph in Figure 14, a spike of highly similar interactions is observed as 1 is approached.

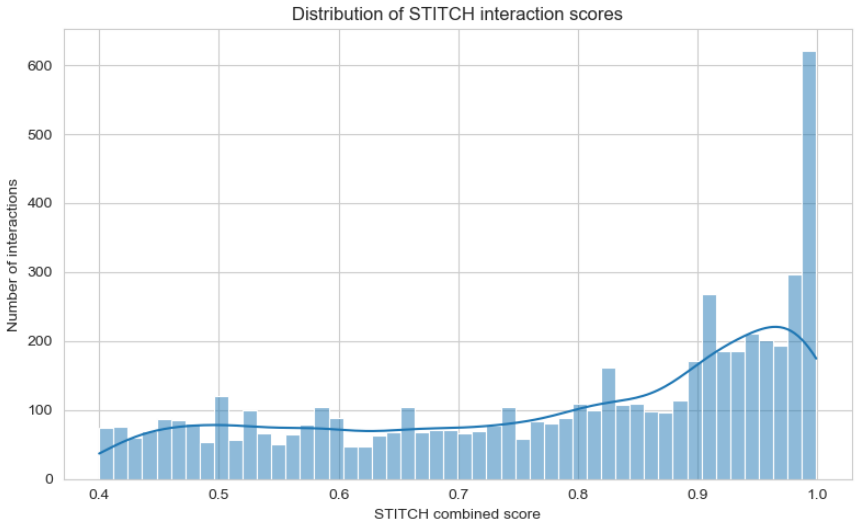

Figure 14. Interaction Score Distribution on network.

In a similar method as in the previous two sections, function `graph_stats_at_threshold()` is used to determine a threshold value to retain only highly similar interaction pairs. For the interaction network, the threshold is determined to be 0.70, in which 67.21% of edges were retained. Table 3.6 contains both unfiltered and filtered graph network statistics.

Attribute	Before Filtering	Threshold >= 0.7
Total Nodes	1157	1082
Total Edges	5585	3754
Density	0.008351	0.00641
Drug Nodes	615	579
Protein Nodes	542	502
Average Degree	9.6542	6.9390

Table 3.6 Drug-Side Effect Network Graph Statistics. Prior to filtering and after filtering at threshold >= 0.7

4 Heterogeneous Graph Construction

The purpose of this section is to describe how the node entities, node attributes, and edge sets were assembled into a heterogeneous graph object for use in relational graph convolutional network (R-GCN) training. The PyTorch Geometric (PyG) library was used to implement the heterogeneous graph construction and representation. The graph schema of *HG* is shown in Table 4.

Source Node	*Relation*	*Destination Node*	*Weighted?*
Drug	*has_se*	*Side Effect*	*No*
Drug	*Semantic_sim*	*Drug*	*Yes (Cosine Similarity)*
Drug	*Chemical_sim*	*Drug*	*Yes (Tanimoto Similarity)*
Protein	*Drug_protein_interaction*	*Drug*	*Yes (STITCH)*
Protein	*Protein_interaction*	*Protein*	*Yes (STITCH)*
Drug	*Drug_Drug_interaction*	*Drug*	*Yes (STITCH)*

Table 4 *HG* schema. Each relation type is stored as a relation-specific edge list using `HG[src_type, rel_type, dst_type.edge_index`, with option for edge weights stored in `edge_attr` for weighted relation.

4.1 Node and Edge Construction

Node features were loaded for each node type using computed feature vectors described in Section 3 that were saved during data preprocessing. For each node type, the list of node identifiers was loaded and mapped to continuous integer indices using an identifier-to-index dictionary to ensure consistent referencing of nodes across relation types. The feature vector was assigned to the heterogeneous graph object *HG* as `HG[node_type].x,` and the total number of nodes was specified using `HG[node_type].num_nodes`. This method was repeated for drug, side effect, and protein nodes.

Relation-specific edge lists were loaded from CSV files with preprocessed edge data, each containing the source node identifier (`src`), destination node identifiers (`dst`) and edge weights (`weight`) if available. The relation attribute (`rel`) was used to distinguish between drug-protein, protein-protein, and drug-drug interaction types in the mixed drug-protein interaction file from the UniProt database. Each row represents a single relationship between two nodes. The node identifiers were mapped to the identifier-to-index dictionaries constructed during node initialization as described previously. The indices were assembled to `edge_index` tensors of shape $[2, |E_r|]$, where 2 denotes the source and destination nodes and $|E_r|$ denotes the number of edges in the relation (e.g.,

`HG['drug', 'has_se', 'se'].edge_index`). If a relation is weighted the similarity or confidence score is stored as `edge_attr` tensor with shape $[|E_r|, 1]$ where 1 denotes the weight of the edge. All relation types were weighted, aside from drug-side effect relation. The tensors were assigned to their relation types in *HG*.

4.2 Conversion to Undirected Graph

With the applied formatting, the graph appears as directed. To allow symmetric message passing between nodes, the heterogeneous graph was converted to an undirected graph representation using graph transformation `ToUndirected()`, which adds reverse edges for each relation type between two different nodes (e.g. `['se', 'rev_has_se', 'drug'].`This produced a bidirectional edge representation for all relations while maintaining relation-specific edge weights if they were available for a given relation.

4.3 PyG HeteroData Representation

After combining adding nodes and edge attributes, the heterogeneous graph was represented with the PyG `HeteroData` object to store the node feature matrices, node counts, and relation-specific edge index and edge attribute tensors for each relation type. The final heterogeneous graph structure is as follows, and confirms successful construction required for R-GCN training.

```
HeteroData(
  se={
    x=[10184, 1268],
    num_nodes=10184,
  },
  drug={
    x=[639, 64],
    num_nodes=639,
  },
  protein={
    x=[502, 345],
    num_nodes=502,
  },
  (drug, has_se, se)={ edge_index=[2, 174977] },
  (drug, semantic_sim, drug)={
    edge_index=[2, 18522],
    edge_attr=[18522, 1],
  },
  (drug, chemical_sim, drug)={
```

```
    edge_index=[2, 2074],
    edge_attr=[2074, 1],
  },
  (drug, drug_interaction, drug)={
    edge_index=[2, 3240],
    edge_attr=[3240, 1],
  },
  (protein, protein_interaction, protein)={
    edge_index=[2, 1082],
    edge_attr=[1082, 1],
  },
  (protein, drug_protein_interaction, drug)={
    edge_index=[2, 1601],
    edge_attr=[1601, 1],
  },
  (se, rev_has_se, drug)={ edge_index=[2, 174977] },
  (drug, rev_drug_protein_interaction, protein)={
    edge_index=[2, 1601],
    edge_attr=[1601, 1],
  }
)
```

5 Relational Graph Neural Network Construction and Training

The purpose of this section is to describe the construction and training of the R-GCN model used to learn embeddings from the heterogeneous graph *HG* and perform drug-side effect link prediction. architecture construction and training. The supervised training procedure using negative sampling and binary classification loss is performed after.

The model is implemented using PyTorch and PyG. The model structure was developed using several learning resources including tutorials (41, 42, 43, 44, 46) and PyG documentation (45). The input is the undirected heterogeneous graph constructed in Section 4.

5.1 R-GCN Architecture

The `RGCNLinkPredictor()`, consists of two components: (i) an R-GCN encoder that generates node embeddings, (ii) a general matrix factorization (GMF) decoder that computes scores between drugs and side effects.

5.1.1 RGCNConv Extension

The R-GCN architecture starts with an extension of the `RGCNConv` class from PyG. The original `RGCNConv` is a relational graph convolution operator, where each relation type is associated with a learnable transformation matrix and updates each node as follows, defined by (22):

$$h_i^{(l+1)} = \sum_{r \in R_E} \sum_{j \in N_r(i)} W_r^{(l)} h_j^{(l)}$$

Where $h_i^{(l+1)}$ is the embedding vector of node *i* at layer *l* , *RE* represents a specific relation in relation set *R,* $N_r(i)$ represents the neighbors of node *i* in relation *r* , $W_r^{(l)}$ is the relation-specific learnable matrix. This equation to update nodes does not include self-loops. If using this implementation, all edges of the same relation type carry an equal importance. However, this implementation fails to incorporate the edge weight attribute, originally represented as a score of similarity or probability of interaction. While the drug-side effect and reverse drug-side effect relation sets are unweighted, the other six relation sets are weighted. Therefore, the `RGCNConv` layer was extended to explicitly incorporate edge weights to the message passing process (`WeightedRGCNConv2`). The node update rule is now:

$$h_i^{(l+1)} = \sum_{r \in R_E} \sum_{j \in N_r(i)} w_{ij} W_r^{(l)} h_j^{(l)}$$

Where w_{ij} is a scalar edge weight. With this extension, the R-GCN model can account for interaction strength or similarity and has more expressive message propagation between nodes of the same type. It is especially suitable for biological data networks, where interaction strength carries quantitative information, and better reflect underlying biological interaction patters and is aimed to improve the quality of learned embeddings, as opposed to not incorporating the edge weights.

5.1.2 R-GCN Encoder Implementation

The `RGCNEncoder` is the message-passing component of the model and is responsible for learning the low-dimensional node embeddings for all node types *D, S, P* in *HG.* The encoder produces relational type-specific embeddings after aggregation across all relational types in R_E. The encoder performs (i) alignment of node feature spaces to a common latent dimension `hidden_channels`, (ii) performs relational message passing, and (iii) returns embeddings split by node type for decoding.

As the node types in heterogeneous graphs have different feature dimensionalities, it is beneficial to first map all node features to a common latent space dimension `hidden_channels=256`. From *HG*, it can be observed that the feature matrix embeddings of each node type is of different dimension:

$$\mathbf{X}^D \in \mathbb{R}^{|D| \times F_D}, \mathbf{X}^S \in \mathbb{R}^{|S| \times F_S}, \mathbf{X}^P \in \mathbb{R}^{|P| \times F_P}$$

Therefore: $$\mathbf{X}^D \in \mathbb{R}^{639 \times 64}, \mathbf{X}^S \in \mathbb{R}^{10184 \times 64}, \mathbf{X}^P \in \mathbb{R}^{502 \times 345}$$

The original side effect feature matrix embedding was of dimension [10184 x 1268]. However, this was later reduced to 64 due to computational cost. The rest of the values can be found in Tables 3.1 – 3.3 in Section 3. To enable relational message passing across all node types, each feature matrix is projected into a shared latent space using node-type-specific linear transformations. This projection is defined as:

$$\mathbf{H}^D = \mathbf{X}^D \mathbf{W}^D, \mathbf{H}^S = \mathbf{X}^S \mathbf{W}^S, \mathbf{H}^P = \mathbf{X}^P \mathbf{W}^P$$

where:

$$\mathbf{W}^D \in \mathbb{R}^{64 \times 256}, \mathbf{W}^S \in \mathbb{R}^{64 \times 256}, \mathbf{W}^P \in \mathbb{R}^{345 \times 256}$$

are learnable projection matrices. These transformations ensure that all node embeddings lie in the same latent space, allowing relational graph convolution to be applied uniformly across drugs, side effects, and proteins.

Although the foundation of this thesis work is the use of heterogeneous graphs to highlight the intricacies of biological and chemical networks, the relational convolution operator `WeightedRGCNConv2` used in this thesis expects a single feature matrix in which all nodes are indexed in a common space. Therefore, the use of the function `_build_homogeneous_x()` constructs this matrix while still retaining the specific node attributes. An empty, unified feature matrix representation: $\mathbf{H}^{(0)} \in \mathbb{R}^{|V| \times 256}$

Where $|V| = |D| + |S| + |P| = 11325$

Is built, then for each node type, the encoder identifies the indices of nodes belonging to that type using a node type tensor, which identifies where each node belongs to the drug, side effect, or protein node set. The corresponding projected feature matrix is inserted to the unified feature matrix to ensure all node embeddings are placed in the correct positions, according to homogeneous graph node ordering. It must also be stated that when converting to a homogeneous graph, the relational structure is maintained in the edge type tensor, which is used during the convolution. Therefore, the multi-relational and multi-nodal structure of the heterogeneous graph is preserved in the homogeneous representation. However, the transformation from heterogeneous graph structure to homogeneous graph structure is computationally inefficient if performed during each forward pass. Thus, as the graph structure is static during training, the homogeneous graph representation is only computed once and cached.

The encoder then constructs a unified weight tensor aligned with the homogeneous graph representation in the `_collect_edge_weights()` function. The output is an assembled, single vector of edge weights corresponding to all edges in the graph. For each relational types in R_E of *HG,* the graph contains an edge attribute tensor $\mathbf{w}^{(r)} \in \mathbb{R}^{|E_r|}$, that represent the weights of edges in that relation *r* and encode similarity scores or interaction scores. The function iterates over relation types and extract edge weights; if no edge weights are found, like in the drug-side effect relation type, a weight of 1.0 is assigned to ensure all edges contribute to message propagation. The individual vectors are concatenated to a single unified edge tensor:

$$\mathrm{W} \in \mathbb{R}^{|\mathrm{E}|}$$

that aligns with the edge ordering of the homogeneous graph representation.

The `forward()` method of `RGCNEncoder` performs one embedding computation and returns the node-type specific embedding matrices of drugs, side effects and proteins as dictionary $E_d =$

out['drug'] $\in \mathbb{R}^{639\times d}$, E_s = out['se'] $\in \mathbb{R}^{10184\times d}$, and E_p = out['protein'] $\in \mathbb{R}^{502\times d}$ where d = `out_channels = 128`. The embeddings E_d and E_s are used as inputs to the decoder, for the main goal of drug-side effect link prediction task. The `forward()` method first checks if homogeneous tensors have been cached (the heterogeneous graph has already been converted to the heterogeneous graph). If so, tensor values are reused to avoid repeating `to_homogeneous()` function. Otherwise, the encoder performs the conversion, along with `_collect_edge_weights()` if parameter `use_edge_weights=True`. The function `_build_homogeneous_x()` is then called to construct the node feature matrix. The core embedding computation occurs by iterating through stacked `WeightedTGCNConv2` layers. For each layer *l* the node representations are updated using the following message passing:

$$H^{(l+1)} = WeightedRGCNConv2(H^{(l)}, edge\ index, edge\ type, edge\ weight)$$

Where *H* represents node representations. The encoder then applies normalization to each embedding to prevent exploding activation, ReLU activation to introduce nonlinearity and allow the model to learn complex pattern, and dropout to prevent overfitting.

After message passing, embeddings for all nodes are stored in homogeneous ordering of dimension [11325, 128]. The encoder reconstructs the type-specific embedding matrices by selecting rows based on node type tensor. The outputs are E_d and E_s for drug embeddings and side effect embeddings, and heterogeneity is restored.

5.1.3 R-GCN Decoder Implementation

To predict the drug-side effect associations from the drug and side effect embeddings learned by `RGCNEncoder()`, a generalized matrix factorization (GMF) decoder is implemented, `GMFDecoder()`. The role of this decoder is to map the drug embeddings E_d and side effect embeddings E_s to a common space to perform matrix factorization and compute a drug-side effect probability score for each pair. $E_d \in \mathbb{R}^{639\times128}$, $E_s \in \mathbb{R}^{10184\times128}$ where d_{emb} = `out_channels = 128`. Although E_d and E_s are in the same embedding dimension of size 128, they are further projected to a lower-dimensional space `hidden_dim = 64`. The projection matrices L_d and R_S are multiplied by E_d and E_s and produce P_d and P_s :

$$P_d = E_d L_d \ , \ P_s = E_s R_s$$

To calculate the final predicted association score between drugs and side effects, the dot product was applied to product B' :

$$B' = P_d P_s^T = (E_d L_d)(E_s R_s)^T$$

Where $B' \in \mathbb{R}^{|D| \times |S|}$

This method follows that of Yu *et al.* (13), in which generalized matrix factorization was used to reconstruct the bi-adjacency matrix, as it allows more flexibility in modeling heterogeneous drug-side effect interactions. In initial training, the dot product magnitudes became excessively large and therefore needed to be scaled. The decoder handles this issue by scaling by the square root of the latent interaction dimension, to improve training stability.

5.1.4 R-GCN Link Prediction Model

The complete link prediction model is implemented in `RGCNLinkPredictor` class, which integrates the R-GCN encoder `RGCNEncoder()` and the R-GCN Decoder `RGCNDecoder()`. The purpose of this module is to compute association scores between drug and side effect nodes using embeddings learned from the heterogeneous graph.

The model follows a two-stage pipeline:

1. Encode nodes using relational graph convolution

2. Decode embeddings to predict drug-side effect associations

The encoder learns embeddings for all nodes in the graph and capture relational information from the heterogeneous graph. The decoder then uses the embeddings to calculate predicted association scores.

The `forward()` method performs a complete prediction pass through the model. The encoder processes the input heterogeneous graph and resulting drug and side effect embeddings are extracted. The embeddings are then passed to the decoder. It is possible to compute score for selected pairs, or for all possible drug-side effect combinations. For this thesis, the latter was initially chosen. The model also provides an `encode()` method that returns the learned drug and side effect embeddings without performing decoding.

The unified architecture of `RGCNLinkPredictor` is an effective implementation of both an encoder and decoder to predict drug side effect associations.

5.2 Training Procedure

Supervised training was utilized for the drug-side effect link prediction task on the bipartite relation (drug, has_se, se) . However, the encoder is designed to perform message passing over the full heterogenous graph structure.

The observed drug-side effect edges were split into training, validation, and test sets using PyG's `RandomLinkSplit` transform. 10% of the labeled edges are held out for validation, another 10% are held out for testing. However, because the graph stores both the forward and reverse edge type to account for the graph being undirected, the splitting is applied to both `[('drug', 'has_se', 'se')]` and `[('se', 'rev_has_se', 'drug')]`. Negative sampling was also chosen at a 1:1 ratio: for each positive drug-side effect association, one unobserved drug-side effect pair is sampled and labeled as negative. The `disjoint_train_ratio` parameter of 0.3 to set aside 30% of edges for supervision, and the remaining 70% for message passing to reduce information leakage. Three `HeteroData` objects of train_data, val_data, and test_data are output and each contain the `edge_label_index` of node pairs to be scored and `edge_label` of positive and sampled negative labels.

Model evaluation was performed using `evaluate()` function. The supervision pairs and corresponding labels are retrieved, and the model outputs the raw, unnormalized score for each pair. The loss is calculated using binary cross entropy with logits `binary_cross_entropy_with_logits(y, y')` where $y \in \{0, 1\}$ and y' is the raw output score, and the output are the probabilities using the sigmoid function. The evaluation function returns AUROC, AUPRC, loss. These and other evaluation metrics will be described in the following Section 6.

Training is performed with the Adam optimizer with L2 regularization (denoted as `weight_decay`). Each epoch proceeds by shuffling labeled supervision pairs, then full batch training where `batch_size=83988`=total number of supervision pairs. The model is implemented and the binary cross entropy with logit loss is computed between predicted score and ground truth labels, then gradients are propagated through the encoder and decoder to update model parameters. The full-batch implementation ensures the gradient update integrates information from all labeled associations and was feasible due to the moderate size of the graph. The gradients are computed by

backpropagation, and to optimize the R-GCN, gradient clipping is applied to rescale gradients if their magnitude becomes excessively large (34).

As mentioned in Section 5.2.1, homogeneous graph caching is utilized to improve the computational efficiency of training so that the heterogeneous to homogeneous graph conversion is performed once with the `precompute_homo()` function, where the `RGCNEncoder` stores the edge index, edge type, node type, and edge weight if available.

6 Results and Discussion

The purpose of this section is to report the performance of `RGCNLinkPredictor` and to evaluate the reconstructed bi-adjacency matrix of drugs and side effects against the original bi-adjacency matrix. This section also aims to answer the following questions:

1. Did `RGCNLinkPredictor` recover observed drug-side effect pairs correctly?

2. Did `RGCNLinkPredictor` assign low predicted probability scores to unobserved drug-side effect pairs correctly?

3. Which unobserved drug-side effect pairs were assigned high predicted probability scores, indicating a novel drug-side effect pair?

6.1 Evaluation Metrics

To assess the performance of the R-GCN link prediction model, evaluation metrics measure the model's ability to distinguish true drug-side effect associations from non-association pairs, and to rank true associations among candidate predictions. Though link prediction can be viewed as a binary classification task, in which the model predicts whether a drug-side effect pair represents a true association, it can also be considered a ranking problem, to return the most likely side effects side effects of a given drug. Therefore, threshold-dependent, threshold-independent, and ranking-based metrics are used to provide a full, comprehensive evaluation of model performance.

The probabilities output as described in Section 5.2 are used to compute both classification and ranking metrics. Model performance was evaluated using both classification and ranking metrics. AUROC, AUPRC, F1 score, and binary cross entropy with logits loss (plotted) measure classification performance, while Mean Reciprocal Rank (MRR) evaluates the ranking quality of predicted drug–side effect associations. The `binary_cross_entropy_with_logits(y, y')` function was implemented as opposed to `binary_cross_entropy`() as it is operated directly on raw model outputs and therefore reduces the numerical instability during optimization. It is also able to handle imbalanced datasets, such as the drug-side effect association data (48). Table 6.1 summarizes the evaluation metrics used, including definition and interpretation.

Metric	*Full Metric Name*	*Description*	*Interpretation*
AUROC	Area Under the Receiver Operating Characteristic Curve	Measures the model's ability to distinguish between positive and negative drug–side effect pairs; evaluates the trade-off between true positive rate and false positive rate	Higher values indicate better discrimination ability; value of 0.5 corresponds to random prediction, while 1.0 indicates perfect classification
AUPRC	Area Under the Precision-Recall Curve	Measures the trade-off between precision and recall across different thresholds	Higher values indicate better performance, with emphasis on correctly identifying true positive associations
MRR	Mean Reciprocal Rank	Measures the ranking quality of predicted side effects. For each drug, it computes the inverse rank of the true side effect among all predicted candidates and averages across all drugs	Higher values indicate that true side effects are ranked closer to the top. A value of 1.0 means the correct association is always ranked first
F1 Score	Harmonic Mean of Precision and Recall	Combines precision and recall into a single metric	Higher values indicate better balance between identifying true positives and minimizing false positives
BCEWithLogits Loss	Binary Cross Entropy with Logits Loss	Measures the difference between predicted score and ground truth labels during training	Lower values indicate better model fit and more accurate predictions

Table 6.1 Evaluation Metrics Used for Drug-Side Effect Link Prediction. Utilized by Yu *et al.* (13).

6.2 Model Training and Convergence

The training behavior of the RGCNLinkPredictor model was evaluated using Binary Cross Entropy with Logits Loss and AUROC across training epochs. After adjusting parameters and hyperparameters, the final set of parameters to train the model were:

- `hidden_channels`: 256
- `out_channels`: 128
- `decoder_hidden`: 64
- `num_layers`: 2

- `dropout`: 0.1
- `lr`: 0.001
- `weight_decay`: 0.001
- `num_epochs`: 300
- `batch_size`: 83988

As observed in Figure 15(a), the training loss decreased steadily through the training process, indicating convergence of the model. Loss initially decreased rapid during the first 40 epochs and began to flatten. The sharp decrease reflects the ability of the model to quickly learn the structural and feature-level patterns from the input network. The following gradual decreasing stabilized below 0.45 by the end of the training. The validation performance was evaluated using AUROC. In Figure 15(b) the rapid increase of AUROC value within the first 40 epochs of early training complements the training loss decrease. This again indicates that the `RGCNLinkPredictor` learned the structural and feature-level patterns from the input network. There was a slight, gradual rise of AUROC score from epoch 50 to 300 and eventually plateaued at around 0.86, which may be interpreted that the model no longer made substantial improvement to predictive ability. The peak AUROC value was reached at epoch 294 with a value of 0.8687.

The convergence behavior and the lack of diverging behavior observed in both training loss and validation AUROC indicates that the model learned the input heterogeneous graph successfully. The stabilization of the validation performance further suggests that the model reached a balance between over and under-fitting, meaning that generalization to unseen drug-side effect associations occurs.

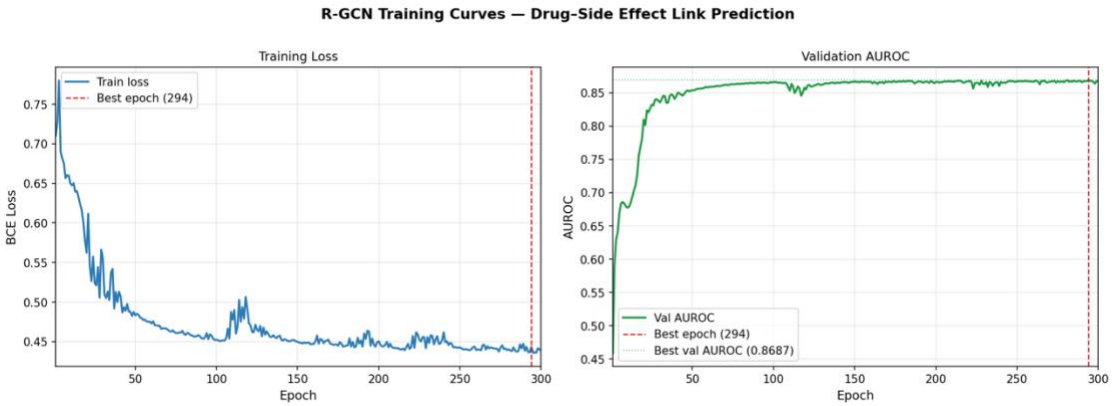

Figure 15(a). Training loss. Loss calculated using `binary_cross_entropy_with_logits()` over 300 epochs. **Figure 15(b). Validation AUROC curve.** Calculated over 300 epochs.

6.3 Link Prediction Performance Evaluation

The predictive performance of the `RGCNLinkPredictor` was further evaluated with the AUPRC, F1 Score, and precision – recall scores. To continue to analyze the model's ability to distinguish between true drug-side effect links and non-associated pairs, Figure 16 displays four visuals containing the following graphs of binary classification metrics: Score Distribution, ROC Curve, Precision-Recall Curve, and Threshold Analysis.

From the top left, Figure 16(a) shows the predicted score distribution for positive and negative drug-side effect pairs. The distribute shows a clear separation between known positive associations (green) and known negative associations (red). The positive pairs correspond to the links in the original dataset are predominantly assigned high predicted probability, as these scores are concentrated closer to 1. The negative pairs correspond to unobserved links in the original dataset are largely lower predicted probabilities of links. The clear separation suggests that the model successfully learned to assign higher confidence to the originally observed drug-side effect links, while assigning lower confidence to unlikely drug-side effect pairs. However, the goal of this thesis was not to reconstruct the original drug side effect matrix, but to reconstruct a matrix with novel drug-side effect pairs. The negative pairs that were assigned moderate to high predicted scores may represent potential unobserved drug-side effect association that have not been documented in the original SIDER database, which will be explored later in the section. There is slight overlap between the distributions; however, this is expected due to the inherent uncertainty of predicting biological associations.

Figure 16(b) of the ROC Curve plots the true positive rate, representing the proportion of correctly identified drug-side effect links, against the false positive rate, which represents the proportion of incorrectly predicted associations among negative pairs. The ROC curve lies above the random diagonal baseline, indicating that the model performs better than random classification of link or no link. The high AUROC value of 0.8741 suggests that the model has a high discriminative performance, and that there is an 87.41% probability that the model assigns a higher predicted score to an observed drug-side effect link than to an unobserved.

The precision-recall curve is another valuable graph to evaluate model performance, especially in the context of a highly imbalanced network. Coupled with the ROC curve, the PR curve provides a more measure of performance. Precision measures the proportion of predicted associations that are correct, while recall measures the proportion of true associations successfully identified by the model. The

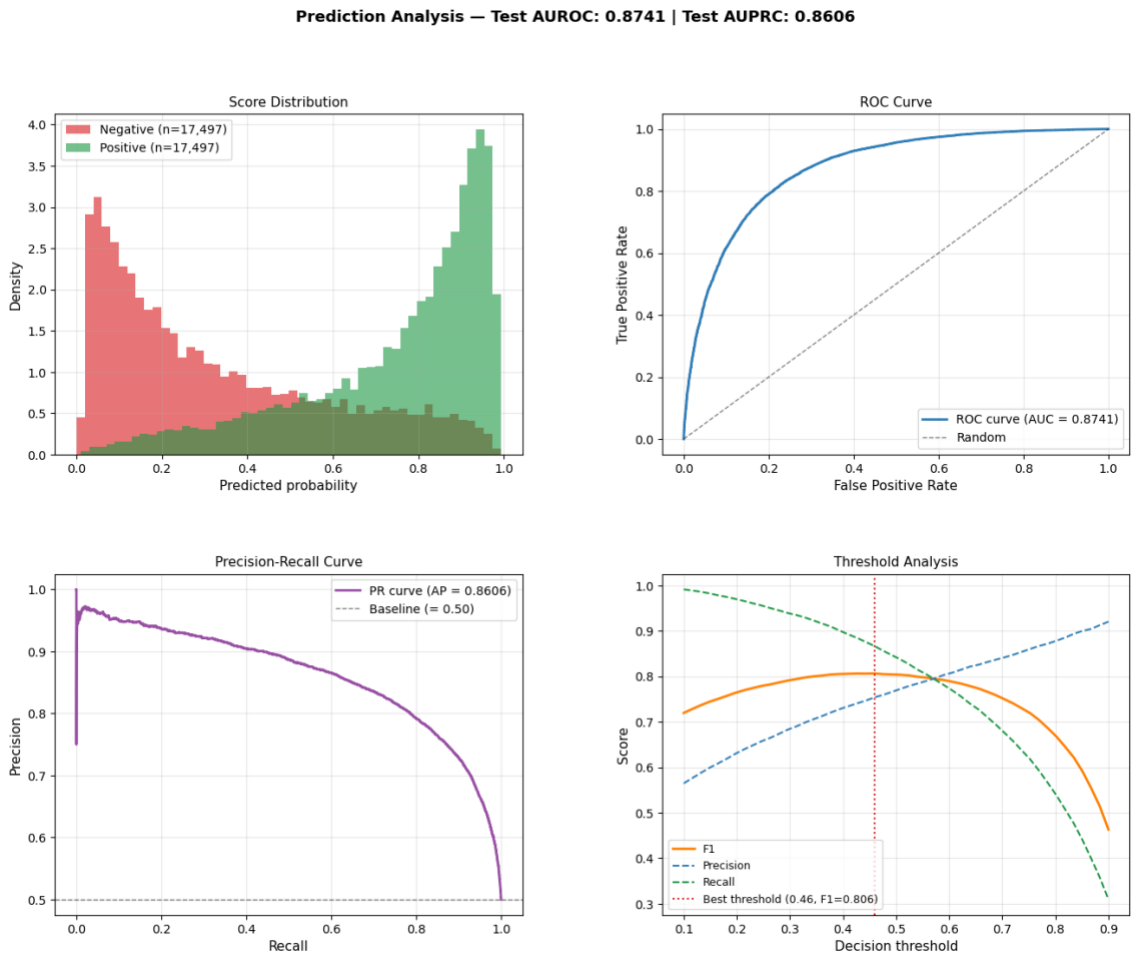

Figure 16. (a) Score Distribution (b) ROC Curve (c) Precision-Recall Curve (d) Threshold Analysis

AUPRC achieved by the model during training was 0.8606, which is yet another indication of strong performance to identify previously observed drug-side effect links.

The Threshold Analysis in Figure 16(d) evaluates the effect of different classification thresholds on model performance and determines the correct decision boundary for distinguishing between drug-side effect links. Precision, recall, and F1-score were computed across a range of classification thresholds from 0 to 1. Precision generally increased as the threshold increased, meaning stricter criteria for predicting association. Conversely, recall decreased as the threshold increased, since stricter thresholds caused the model to miss some true associations. The F1-score, which represents the harmonic mean of precision and recall, reached its maximum value of approximately 0.806 at a decision threshold of approximately 0.46. Therefore, drug-side effect pairs with a predicted probability score above the optimal decision threshold are likely to be true pairs.

6.4 Adjacency Matrix Reconstruction

The central objective of the RGCNLinkPredictor framework is to reconstruct the drug-side effect bi-adjacency matrix using the relational node embeddings learned from the heterogeneous graph input. The original bi-adjacency matrix *B* is binary and highly sparse (see Figure 7). In contrast, the reconstructed bi-adjacency matrix *B'* contains continuous values between 0 and 1 and represent the predicted probability of a link between each drug-side effect pair. From the previous analysis regarding decision threshold, it can be inferred that drug-side effect pairs with a score above 0.46 may be either true links or novel positive links.

Figure 17 illustrates *B'* for the purpose of showing the complete, filled matrix. Recall the sparsity of bi-adjacency matrix *B* of 2.69 % (174,977 positive links): the sparsity of the reconstructed bi-adjacency matrix *B'* is 29.62% (1,927,529 positive links). The drugs and side effects are ordered in the same manner as in the original matrix. Though it is difficult to distinguish the probability value in each individual cell, other observations may be made. Just as in Figure 7, there are strong, dark bands along the x-axis and y-axis that are indicative of drugs that are associated with many side effects, and side effects that occur from drugs respectively. Considering the model performance, it appears that the bands observed in *B* are much more distinct in *B',* reinforcing that the model was able to correctly assign the observed drug-side effect pairs higher predicted probabilities. A stark light band along the y axis in *B'* indicates that a group of side effects have low predicted probability of being associated with drugs and may be rare side effects.

Figure 18 shows a sample of 200 drugs and 500 side effects. The difference in the sparse bi-adjacency matrix *B* versus dense reconstructed bi-adjacency matrix *B'* is even clearer. In this visual, as in Figure 17, broken "bands" of a common side effects at x = 490 of matrix *B* appears to be completely solidified at x=490 of matrix *B'*, indicating that the side effect is predicted to be associated with even more drugs, likely due to its commonality in the original matrix. However, the reconstruction matrix reveals latent banding that cannot be observed in the original matrix, such as the drug at y= 165 that is predicted to have high probability of association with many side effects.

6.5 Analysis of Known vs Unknown Associations

A statistical analysis was performed on the predicted probability scores assigned to the observed and unobserved drug-side effect pairs. As outlined in the previous subsection, the reconstructed matrix *B'*

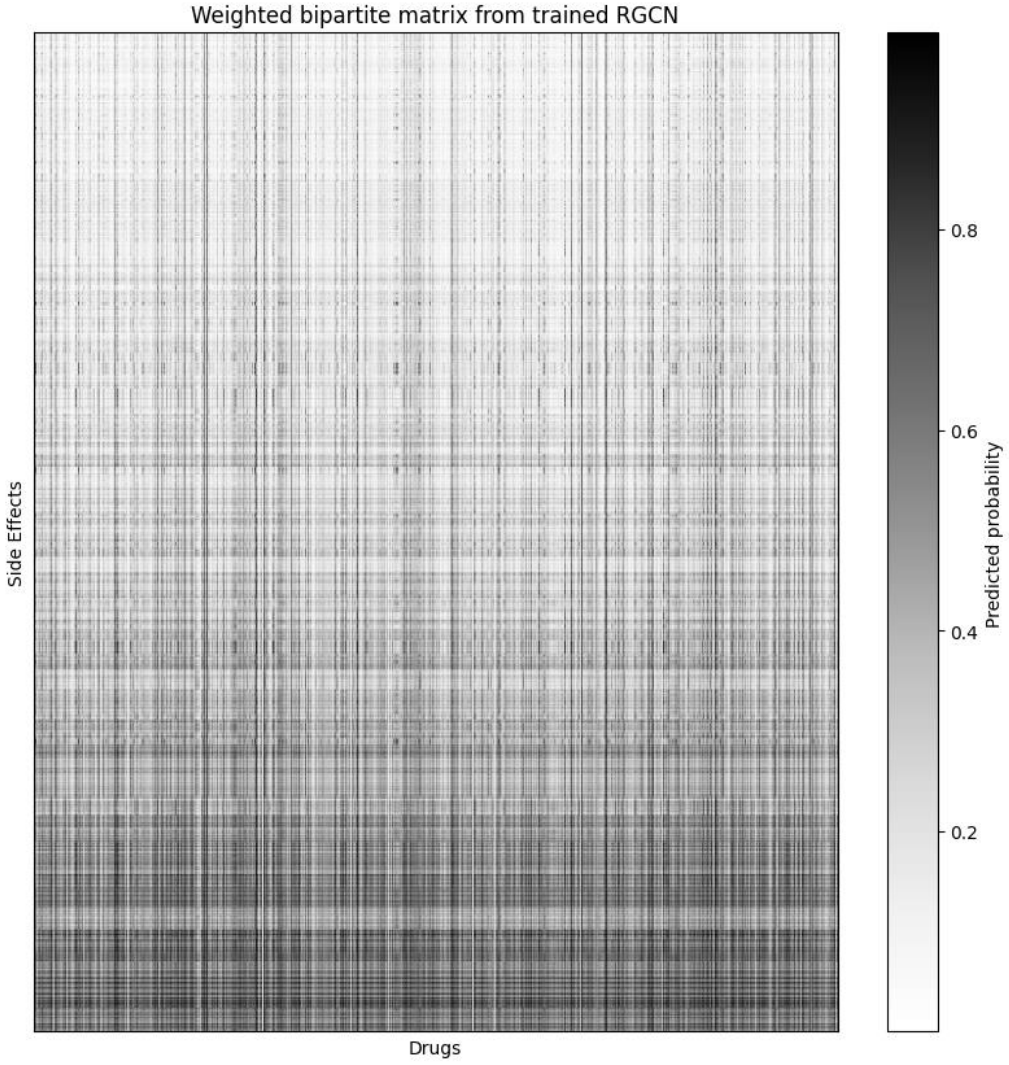

Figure 17. Full reconstructed bi-adjacency matrix *B'*.

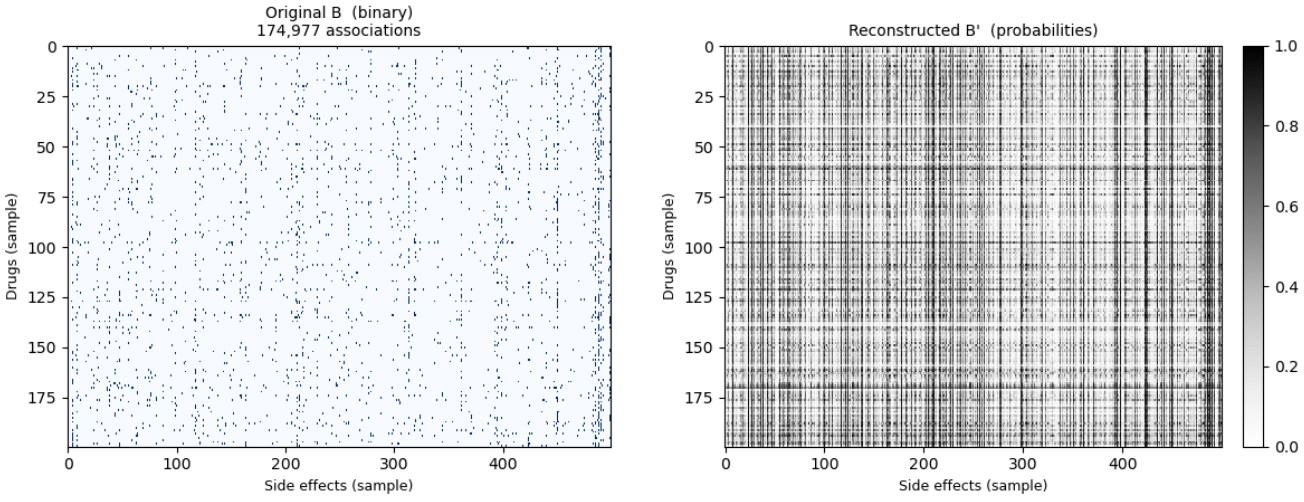

Figure 18. Sample of original bi-adjacency matrix *B* and reconstructed *B'*

contains all predicted association probabilities for all possible drug-side effect pairs, and can be directly compared to the values in bi-adjacency matrix *B.* The analysis was performed by returning a variable containing all drug side effect pair predicted probabilities in *B'* where drug-side effect pair in *B =1*, (`pos_score`) and a separate variable containing those in *B'* where *B*=0 (`neg_score`). The analysis revealed quantitatively that known drug-side effect associations received higher predicted probabilities, with a median score of 0.8042 indicating that half of observed drug-side effect pairs have a predicted probability of 80.42%. By contrast, drug-side effects pairs that were not originally observed received lower predicted score, with the median score equal to 0.2376 indicating that half of unknown pairs receive probability scores below 23.76% and the model is rejecting unobserved drug-side effect pairs. Furthermore, the fraction of drug-side effect pairs below predicted probability of 0.5 is 75.72%, indicating the model assigns over three-quarters of unobserved drug-side effect pairs low probabilities.

The mean score separation is a value that is calculated by subtracting the mean negative score from the mean positive score: 0.7369-0.3189 = 0.4182 . This value represents the discrimination ability of the model; in this case true associations receive a score 0.4180 higher on average. However, some observed drugs and side effect pairs received a low predicted probability score, such as the minimum of 0.0002. This may be due to sparse connectivity in the input graph. This analysis quantitatively confirms the model's ability to distinguish known and unknown drug side effects.

6.6 Identification of Novel Drug-Side Effect Associations

The primary aim of this work is to predict previously unobserved drug-side effect associations. To evaluate this capability, the predicted associations were examined for drug-side effect pairs not included in the original adjacency matrix. As mentioned in Section 6.4 , the model identified 1,927,529 positive links between drugs and side effects. Upon further analysis, it was determined that 92.11% of those links are novel pairs, such that 1,775,456 new drug-side effect pairs were predicted. Figure 19 shows the distribution of predicted probability scores of novel pairs, which exhibits a decreasing trend from the decision threshold toward higher predicted probability values. The majority of novel predictions fall within 0.5 to 0.7, indicating moderate confidence, with fewer predictions receiving high confidence scores above 0.9. Table 6.1 presents the top 20 highest-confidence novel drug-side effect pairs ranked by probability score. The predicted scores for these novel associations were consistently high, with values exceeding 0.99. Several drugs appeared

multiple times, including Chlordiazepoxide, Meperidine, and Flurazepam, suggesting that the model identified consistent patterns linking these drugs to specific side effects. The recurrence of the

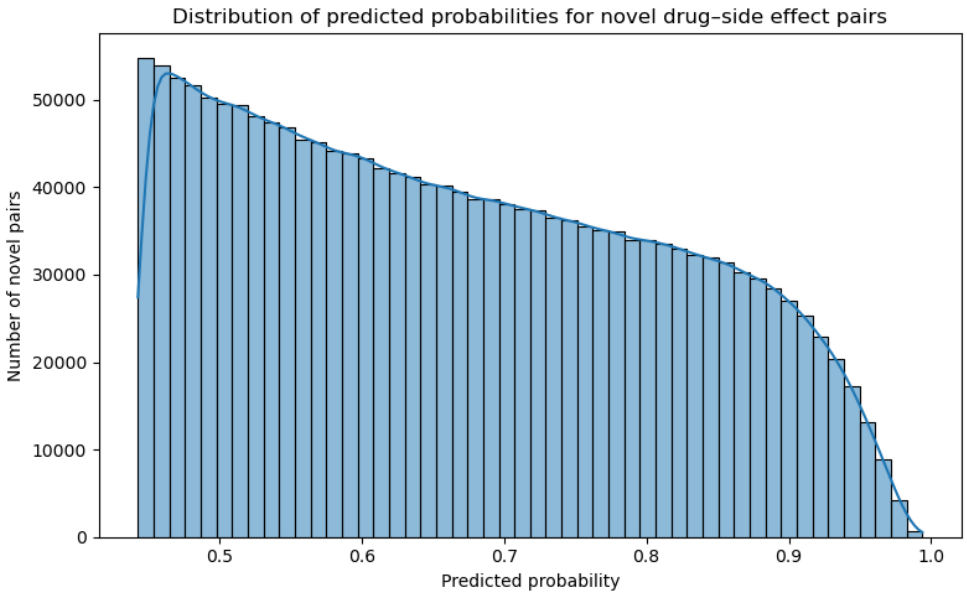

Figure 19. Novel Drug-Side Effect Predicted Probability Score Distribution.

specific side effects including ‘oxygen saturation decreased’ and ‘hepatic enzyme’ solidify this inference. While these predicted associations require experimental or clinical validation, for the purpose of the thesis, they represent biologically ideal candidates for further investigation, if indeed these drugs cause these side effects. This capability shows the potential of heterogeneous networks and R-GCN implementation to be a powerful mechanism for adverse drug reaction discovery.

The drugs with the most novel predicted side effect can be seen in Figure 20. The model predicted that Retinoic acid, leflunomide, clozapine and deferasirox gained over 8000 novel side effects. A similar analysis was performed on the most frequently predicted novel side effects: ‘infusion site pruritus’, ‘pharyngolaryngeal discomfort’, and ‘hypertonia neonatal’ were found to be the top three most novel predicted side effects, being associated 625, 624, and 624 times, respectively. Lastly, the ATC level 1 and level 2 distributions by novel pair prediction was analyzed. The majority of novel predicted drug–side effect associations were observed in ATC level 1 classes N (n=349,155), L (n=237,909), C (n=233,435), and J (n=188,082), indicating that nervous system, antineoplastic, cardiovascular, and anti-infective drugs contributed the largest number of predicted novel side effects, being associated 625, 624, and 624 times, respectively. Lastly, the ATC level 1 and level 2 distributions by novel pair prediction was analyzed. The majority of novel predicted drug–side effect

Rank	*Drug Name*	*Side Effect*	*Score*
1	Chlordiazepoxide	oxygen saturation decreased	0.9952
2	Flurazepam	oxygen saturation decreased	0.9946
3	Meperidine	hepatic enzyme	0.9945
4	Triazolam	oxygen saturation decreased	0.9945
5	Chlordiazepoxide	hepatic enzyme	0.9939
6	Meperidine	blood potassium decreased	0.9939
7	Prochlorperazine	oxygen saturation decreased	0.9938
8	Calcium	oxygen saturation decreased	0.9937
9	Chlordiazepoxide	cyanosis	0.9937
10	Methylprednisolone	oxygen saturation decreased	0.9936
11	Chlordiazepoxide	blood potassium decreased	0.9935
12	Tobramycin	metastases to liver	0.9934
13	Lorazepam	cyanosis	0.9934
14	Alprazolam	oxygen saturation decreased	0.9934
15	Triazolam	hepatic enzyme	0.9933
16	Flurazepam	amphetamines	0.9932
17	Triazolam	musculoskeletal stiffness	0.9932
18	Chlorhexidine	blood alkaline phosphatase increased	0.9931
19	Triazolam	blood potassium decreased	0.9931
20	Chlordiazepoxide	dilation atrial	0.9931

Table 6.1 Novel Drug-Side Effect pairs

associations were observed in ATC level 1 classes N (n=349,155), L (n=237,909), C (n=233,435), and J (n=188,082), indicating that nervous system, antineoplastic, cardiovascular, and anti-infective drugs contributed the largest number of predicted novel side effects, which may be indicative of these categories have broader or less characterized side effect profiles. Drilling down to the ATC level 2 classification distribution, the highest number of novel predictions occurred among drugs lacking ATC Level 2 classification (n=176,345), followed by antineoplastic agents (L01), psycholeptics (N05), antibacterials (J01), and psychoanaleptics (N06). However, it must be noted that these four ATC level 2 classes were among those with the largest number of drugs in the original data set (see Figure 10), suggesting that the frequency of novel predictions is partly influenced by class size.

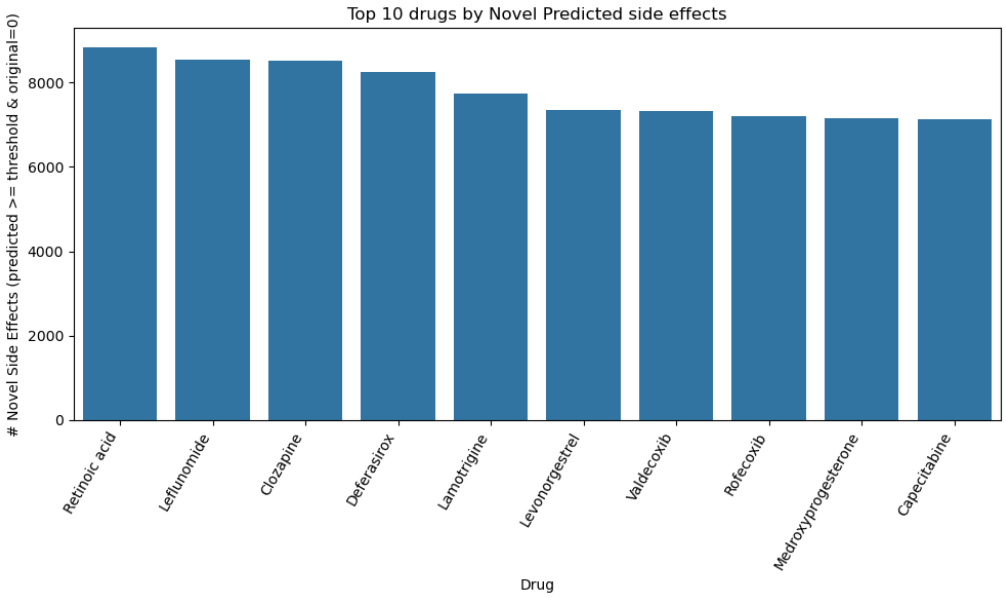

Figure 20. Top 10 drugs by novel predicted side effects.

6.7 Hyperparameter Optimization

The final analysis performed on the model is the evaluation of the impact of regularization parameters including dropout rate and L2 weight decay on model performance. Both dropout and L2 regularization influence the model's ability to generalize by controlling overfitting and stabilizing the training of the neural network. The model performance was relatively stable across different dropout values, with the best validation AUPRC achieved at a dropout rate of 0.001. Lower dropout values resulted in slightly reduced generalization performance, while excessive dropout may limit the model's ability to fully utilize learned relational representations. Overall, the results indicate that moderate regularization improves model generalization slightly while maintaining strong predictive performance.

Similarly, the effect of L2 weight decay and dropout on Mean Reciprocal Rank (MRR) was evaluated. The MRR values remained consistently low across different weight decay and dropout settings, with minimal sensitivity to the dropout or L2 hyperparameters. expected given the highly sparse nature of the drug-side effect adjacency matrix, where the number of possible associations is extremely large relative to the number of true associations. Therefore, it is justified to say that in large, sparse biomedical networks, metrics such as AUROC and AUPRC provide a more reliable measure of overall model discrimination performance, while MRR may underestimate model effectiveness due to the large candidate space. Nevertheless, the stability of performance across

hyperparameter settings demonstrates that the weighted R-GCN framework is robust and does not rely heavily on precise hyperparameter tuning to achieve strong predictive performance.

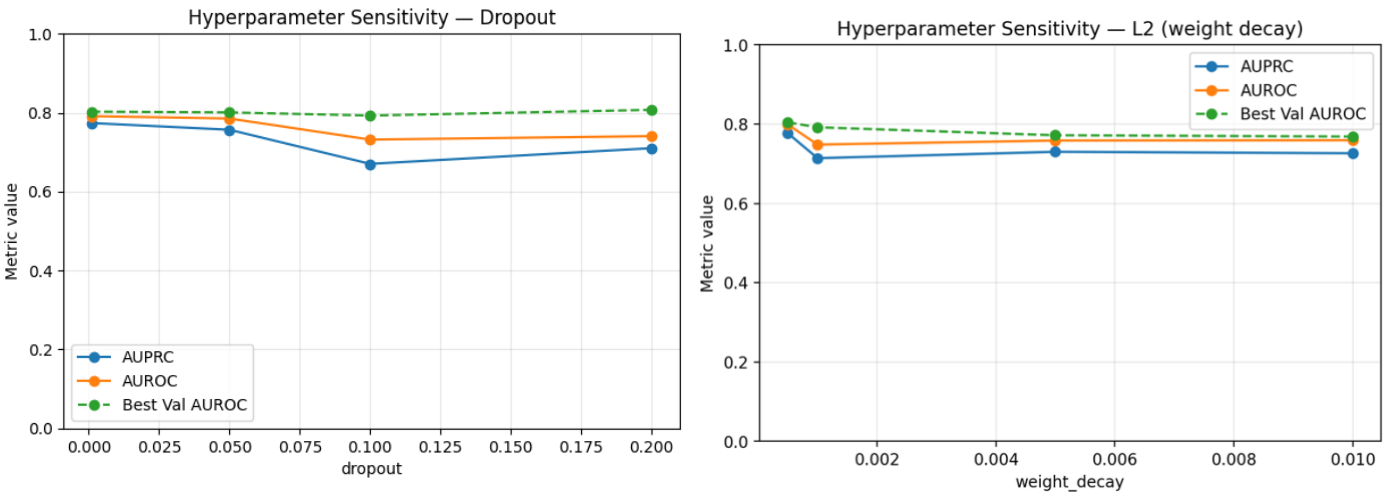

Figure 21. Dropout and L2 weight decay grid search. Performance of the model did not change significantly.

7 Conclusion and Future Directions

This thesis addresses the issue of predicting unknown adverse drug reactions (side effects) by modeling drug-side effect relationships as a structural link prediction task using a heterogeneous graph framework. Traditional ADR detection methods rely heavily on clinical trials and reporting systems, which have the limitations of high cost and reporting bias. This thesis proposed a computational approach that integrates chemical, semantic, and molecular and biological interaction data to a heterogeneous network representation and applies Relational Graph Convolution Network framework to learn the predicted probabilities between drug and side effect links. The framework combined and transformed several biomedical data sources, including known drug-side effect associations, drug-drug semantic similarity, chemical structure, drug-protein interactions and protein-protein interactions. The combination of these data sources to the heterogeneous graph structure with PyG `HeteroData()` object allows for the use of a more comprehensive, complete representation of latent mechanisms that contribute to adverse drug reactions. In biological systems, drugs and proteins interact with each other and themselves through complex and interdependent pathways, as opposed to isolated interactions and therefore the heterogeneous graph representation is utilized to preserve the realistic relational structures by modeling different entity types and relation types to ultimately improve potential links between drug-side effect associations. The use of the R-GCN takes advantage of the multi-relational nature of the heterogeneous network, such that the model differentiated between biological interaction relationship, semantic and chemical similarity relationships and iteratively aggregated neighbor information of node embeddings, reflecting the structural connectivity of the graph and node feature representations. It is also worth mentioning that the framework is suited for both small and large-scale networks, especially in the case that validating all possible drug-side effect associations is not possible. This approach contributes to improving the computation drug safety prediction and demonstrates the effectiveness of heterogeneous graph networks to model biological systems.

The experimental results demonstrate that the proposed weighted R-GCN framework effectively learned meaningful relational representations from the heterogeneous biomedical network and achieved strong predictive performance, with a test AUROC of 0.8640 and AUPRC of 0.8567. The model successfully reconstructed the sparse drug–side effect adjacency matrix and assigned significantly higher predicted probabilities to known associations compared to unknown pairs, confirming its ability to distinguish true biological relationships, as reported in the drug-side effect dataset from SIDER. Furthermore, the identification of high-confidence novel drug–side effect

predictions highlights the model's capability to generalize beyond the observed data and infer new drug-side effect associations, placing an emphasis on its potential utility in computational drug analysis.

Limitations of this thesis primarily relate to data completeness and model design constraints. The reliance on publicly available databases may lead to incomplete or biased information. For example, the SIDER database in which the drug-side effect network was obtained from extracts information from adverse reporting systems and public documents, which may not incorporate true associations that are unreported. Additionally, to increase computational efficiency of the model, it was necessary to impose thresholds so that not all computed drug similarity and drug-protein/protein-protein/drug-drug interactions were included in the final heterogeneous graph. Therefore, the heterogeneous graph inherently does not represent a *complete* biological interaction landscape. The sparsity of the drug-side effect network and imbalance of positive and negative samples in the set. As observed in the exploratory data analysis, the drug-side effect bi-adjacency matrix is highly sparse in that the number of negative samples exceeds positive samples (drug-side effect link). It may be possible that some negative pair samples represent unknown true associations between drugs and side effects, which may increase noise in training. The feature representation choices made in the thesis work may have also presented limitations due to the dimensionality reduction that was applied to improve computational efficiency. The reduction of dimensionality may have resulted in partial loss of information contained in the original embeddings. The model architecture itself introduces limitations. Although the extension of PyG RGCNConv to account for edge weights (WeightedRGCNConv2) such that similarity scores and interaction probabilities may influence message passing, the model does not implement attention-based mechanisms to dynamically learn the importance of neighbors during training. The future use of attention-based GCNs may be of interest to learn adaptive edge weight, as opposed to the fixed edge weights in the implemented model, and neighbor importances. Additionally, the depth of the R-GCN network and embedding dimensionality were constrained by available computational resources on the local machine.

While this work does indeed demonstrate the feasibility of predicting unseen drug-side effect associations using multi-relational biological and biomedical data, future work can be performed to improve the model. As mentioned, the use of an attention-based graph neural network architecture such as Relational-Graph Attention Networks (RGATs), which extend attention mechanisms to relational graphs as studied by Busbridge *et al.* (47), to learn the relative importance of neighbors and relation types so that the model may prioritize the most informative interactions with the

potential to improve prediction accuracy. Another obvious improvement would be to expand the heterogeneous graph with additional sources to richen node attributes and provide additional context to the mechanisms that may cause adverse drug reactions. Finally, future work could explore different applications of this framework, such as drug-target interaction prediction or disease-gene association prediction. The constructed framework is highly flexible and can be applied to various link prediction tasks involving biological data.

This thesis contributes to a newer field of graph-based machine learning in bioinformatics by demonstrating how R-GCNs can address link prediction tasks. As available biomedical and biological data increases in scale and complexity, this approach is a step in the direction for supporting data-driven healthcare research, drug discovery, and improving upon current ADR detection methods.

References

1. Kommu, S., Carter, C., & Whitfield, P. (2024). *Adverse drug reactions*. In StatPearls. StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK599521/
2. International drug monitoring: the role of national centres. Report of a WHO meeting. (1972). *World Health Organization technical report series*, *498*, 1–25.
3. Edwards, I. R., & Aronson, J. K. (2000). Adverse drug reactions: definitions, diagnosis, and management. *Lancet (London, England)*, *356*(9237), 1255–1259. https://doi.org/10.1016/S0140-6736(00)02799-9
4. Ahmed Amer Mousa Alqarny, Saleh Mousa Balgeeth Alqrni, Saber Ali Atiah Alqarni, Salem Atif Ali Alshehri, Ali Suliman Mohammed Alshehri, & Hassan Abdullah Ali Al Shehri, Sultan Amer Ahmed Al Barqi ,Mohammad Saad Alshehri, Ali Thabet Mgbel Alshehri, Faiz Ali Gaber Al-Shehry. (2024). Adverse Drug Reactions Detection and Management Strategies Overview. *Journal of International Crisis and Risk Communication Research* , 2356–2376. https://doi.org/10.63278/jicrcr.vi.1232
5. Whitebread, S., Hamon, J., Bojanic, D., & Urban, L. (2005). Keynote review: in vitro safety pharmacology profiling: an essential tool for successful drug development. *Drug discovery today*, *10*(21), 1421–1433. https://doi.org/10.1016/S1359-6446(05)03632-9
6. FDA. (2024). *FDA's Adverse Event Reporting System (FAERS).* U.S. Food and Drug Administration. https://www.fda.gov/drugs/surveillance/fdas-adverse-event-reporting-system-faers
7. Kuhn, M., Letunic, I., Jensen, L. J., & Bork, P. (2016). The SIDER database of drugs and side effects. *Nucleic acids research*, *44*(D1), D1075–D1079. https://doi.org/10.1093/nar/gkv1075
8. Kuang, Q., Wang, M., Li, R., Dong, Y., Li, Y., & Li, M. (2014). A Systematic Investigation of Computation Models for Predicting Adverse Drug Reactions (ADRs). *PLoS ONE*, *9*(9), e105889. https://doi.org/10.1371/journal.pone.0105889
9. Huang, L.-C., Wu, X., & Chen, J. Y. (2013). Predicting adverse drug reaction profiles by integrating protein interaction networks with drug structures. *PROTEOMICS*, *13*(2), 313–324. https://doi.org/10.1002/pmic.201200337
10. Dey, S., Luo, H., Fokoue, A., Hu, J., & Shang, P. (2018). Predicting Adverse Drug Reactions Through Interpretable Deep Learning Framework. *BMC Bioinformatics, 19(Suppl 21),* 476. https://doi.org/10.1186/s12859-018-2544-0

11. Timilsina, M., Tandan, M., d'Aquin, M., & Yang, H. (2019). Discovering Links Between Side Effects and Drugs Using a Diffusion Based Method. *Scientific Reports*, *9*(1). https://doi.org/10.1038/s41598-019-46939-6
12. Kuhn, M., von Mering, C., Campillos, M., Jensen, L. J., & Bork, P. (2008). STITCH: interaction networks of chemicals and proteins. *Nucleic acids research*, *36*(Database issue), D684–D688. https://doi.org/10.1093/nar/gkm795
13. Yu, L., Cheng, M., Qiu, W., Xiao, X., & Lin, W. (2022). idse-HE: Hybrid embedding graph neural network for drug side effects prediction. *Journal of biomedical informatics*, *131*, 104098. https://doi.org/10.1016/j.jbi.2022.104098
14. Aditya Krishna Menon, & Elkan, C. (2011). Link Prediction via Matrix Factorization. *Lecture Notes in Computer Science*, 437–452. https://doi.org/10.1007/978-3-642-23783-6_28
15. Kumar, A., Singh, S. S., Singh, K., & Biswas, B. (2020). Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and Its Applications*, *553*, 124289. https://doi.org/10.1016/j.physa.2020.124289
16. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). *Neural Collaborative Filtering*. https://doi.org/10.48550/arxiv.1708.05031
17. *Adjacency matrix*. (2021, October 4). Wikipedia. https://en.wikipedia.org/wiki/Adjacency_matrix
18. Yu, L., Qiu, W., Lin, W., Cheng, X., Xiao, X., & Dai, J. (2022). HGDTI: predicting drug–target interaction by using information aggregation based on heterogeneous graph neural network. *BMC Bioinformatics*, *23*(1). https://doi.org/10.1186/s12859-022-04655-5
19. Zhang, X. M., Liang, L., Liu, L., & Tang, M. J. (2021). Graph Neural Networks and Their Current Applications in Bioinformatics. *Frontiers in genetics*, *12*, 690049. https://doi.org/10.3389/fgene.2021.690049
20. Zhang, M., & Chen, Y. (2018). Link Prediction Based on Graph Neural Networks. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1802.09691
21. Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1609.02907

22. Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2018). Modeling Relational Data with Graph Convolutional Networks. *The Semantic Web*, 593–607. https://doi.org/10.1007/978-3-319-93417-4_38
23. Thanapalasingam, T., van Berkel, L., Bloem, P., & Groth, P. (2022). Relational graph convolutional networks: a closer look. *PeerJ Computer Science*, *8*, e1073. https://doi.org/10.7717/peerj-cs.1073
24. Zitnik, M., Sosič, R., Maheshwari, S., & Leskovec, J. (2018, August). *BioSNAP datasets: Stanford biomedical network dataset collection*. Stanford University. http://snap.stanford.edu/biodata
25. Kim, S., Chen, J., Cheng T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., Zaslavsky, L., Zhang, J., & Bolton, E. E. (2025). PubChem 2025 update. *Nucleic Acids Res., 53(*D1), D1516-D1525. https://doi.org/10.1093/nar/gkae1059
26. The UniProt Consortium. (2025). UniProt: The universal protein knowledgebase in 2025. *Nucleic Acids Research, 53*(D1), D609–D617. https://doi.org/10.1093/nar/gkae1010
27. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics, 36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682
28. RDKit: Open-Source Cheminformatics Software. https://www.rdkit.org.
29. Beam, A. L., Kompa, B., Fried, I., Palmer, N. P., Shi, X., Cai, T., & Kohane, I. S. (2018). Clinical concept embeddings learned from massive sources of medical data. *arXiv*. https://arxiv.org/abs/1804.01486
30. NVIDIA Corporation. (2024). *ESM-2 model documentation*. BioNeMo Framework. https://docs.nvidia.com/bionemo-framework/2.0/models/esm2/
31. Wikipedia Contributors. (2024, August 6). *Simplified Molecular Input Line Entry System*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Simplified_Molecular_Input_Line_Entry_System
32. Dablander, M. (2022, November 29). *How to turn a SMILES string into an extended-connectivity fingerprint using RDKit | Oxford Protein Informatics Group*. Blopig.com.

https://www.blopig.com/blog/2022/11/how-to-turn-a-smiles-string-into-an-extended-connectivity-fingerprint-using-rdkit/

33. *ATCDDD - Home*. (n.d.). Atcddd.fhi.no. https://atcddd.fhi.no/
34. Thor, W.-M. (2026). *Optimization Strategies for GNNs*. Apxml.com; ApX Machine Learning. https://apxml.com/courses/graph-neural-networks-gnns/chapter-3-gnn-training-complexities/gnn-optimization-strategies
35. *Louvain method*. (2022, June 27). Wikipedia. https://en.wikipedia.org/wiki/Louvain_method
36. *Louvain — scikit-network 0.33.0 documentation*. (2020). Readthedocs.io. https://scikit-network.readthedocs.io/en/latest/tutorials/clustering/louvain.html
37. Zoehler, B. (2024, December 12). *Molecule Clustering Using the Butina Algorithm: Application with Python and RDKit*. Medium. https://zoehlerbz.medium.com/molecule-clustering-using-the-butina-algorithm-application-with-python-and-rdkit-ffb5b5721447
38. Dhimansakshi. (2025, November 25). *GitHub Repository:* Medium. https://medium.com/@dhimansakshi917/molecular-clustering-and-similarity-search-using-rdkit-and-python-f6e3e1d59354
39. Wikipedia Contributors. (2024, October 13). *β-Lactam*. Wikipedia; Wikimedia Foundation.
40. *Cephalosporin*. (2020, May 2). Wikipedia. https://en.wikipedia.org/wiki/Cephalosporin
41. Ribeiro, A. (2022). *Lectures – Graph Neural Networks*. Upenn.edu. https://gnn.seas.upenn.edu/lectures/
42. pyg-team. (2025). *pytorch_geometric/examples/rgcn.py at master · pyg-team/pytorch_geometric*. GitHub. https://github.com/pyg-team/pytorch_geometric/blob/master/examples/rgcn.py
43. Octavian.ai. (n.d.). Machine learning on graphs course. Google Colab. https://colab.research.google.com/drive/1inLXGGbxA3-icR4gI0KdSe47GG2WxetW?authuser=0
44. Volzhin, N. (2025, June). *Graph Convolutional Networks (GCN): All You Need to Know & Code Implementation*. Medium. https://medium.com/@volzhinnv/graph-convolutional-networks-gcn-all-you-need-to-know-code-implementation-fdfcde657b5c

45. *torch_geometric.nn.conv.RGCNConv — pytorch_geometric documentation*. (2024). Readthedocs.io. https://pytorch-geometric.readthedocs.io/en/2.6.1/generated/torch_geometric.nn.conv.RGCNConv.html
46. Loos, T. (2022, November 15). *Entity Type Prediction with Relational Graph Convolutional Network (PyTorch)*. Medium; TDS Archive. https://medium.com/data-science/setup-for-entity-type-prediction-with-relational-graph-convolutional-network-pytorch-3554be0bfd5a
47. Busbridge, D., Sherburn, D., Cavallo, P., & Hammerla, N. Y. (2019). Relational Graph Attention Networks. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1904.05811
48. Sahilcarterr. (2023, December 26). *Why nn.BCEWithLogitsLoss Numerically Stable*. Medium. https://medium.com/@sahilcarterr/why-nn-bcewithlogitsloss-numerically-stable-6a04f3052967