DRUG-DRUG INTERACTION PREDICTION USING KNOWLEDGE GRAPH NEURAL NETWORKS

PROJECT REPORT

SUBMITTED TO

SVKM'S NMIMS (Deemed to be) UNIVERSITY

IN PARTIAL FULFILLMENT FOR THE DEGREE OF

MASTER OF SCIENCE
IN
DATA SCIENCE

BY

NIDHI DAHIYA

NMIMS NILKAMAL SCHOOL OF MATHEMATICS,
APPLIED STATISTICS & ANALYTICS
V. L. Mehta Road, Vile- Parle (West)
Mumbai – 400056

APRIL 2025

M. Sc. Data Science

2025

DRUG-DRUG INTERACTION PREDICTION USING KNOWLEDGE GRAPH NEURAL NETWORKS

PROJECT REPORT

SUBMITTED TO

SVKM'S NMIMS (Deemed to be) UNIVERSITY

IN PARTIAL FULFILLMENT FOR THE DEGREE OF

MASTER OF SCIENCE
IN
DATA SCIENCE

BY

NIDHI DAHIYA

NMIMS NILKAMAL SCHOOL OF MATHEMATICS,
APPLIED STATISTICS & ANALYTICS

V. L. Mehta Road, Vile- Parle (West)
Mumbai – 400056

APRIL 2025

DRUG-DRUG INTERACTION PREDICTION USING KNOWLEDGE GRAPH NEURAL NETWORKS

PROJECT REPORT

SUBMITTED TO

SVKM'S NMIMS (Deemed to be) UNIVERSITY

IN PARTIAL FULFILLMENT FOR THE DEGREE OF

MASTER OF SCIENCE
IN
DATA SCIENCE

BY

NIDHI DAHIYA

Program Chairperson
(Dr. Kavita Jain)

Dean NSOMASA
(Dr. Sushil Kulkarni)

NMIMS NILKAMAL SCHOOL OF MATHEMATICS,
APPLIED STATISTICS & ANALYTICS
V. L. Mehta Road, Vile- Parle (West)
Mumbai – 400056

CERTIFICATE

This is to certify that work described in this thesis entitled “DDI Prediction using KGNN” has been carried out by Nidhi Dahiya under my supervision. I certify that this is his/her bonafide work. The work described is original and has not been submitted for any degree to this or any other University.

Date: 15-04-2025

Place: Mumbai

Internal Mentor
(Dr. Leena Kulkarni)

Date: 15-04-2025

NILKAMAL SCHOOL OF MATHEMATICS,
APPLIED STATISTICS & ANALYTICS

NMIMS NILKAMAL SCHOOL OF MATHEMATICS,
APPLIED STATISTICS & ANALYTICS
V. L. Mehta Road, Vile- Parle (West)
Mumbai – 400056

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Leena and Professor Kavita, our esteemed project mentors, for their unwavering support, expert guidance, and constructive feedback throughout the course of this project.

I am also thankful to the faculty members and academic coordinators at NSOMASA for fostering an environment of academic excellence and providing the resources necessary to undertake this project. The knowledge imparted throughout the program has significantly contributed to the successful execution of this study.

I extend my appreciation to my team member, whose collaboration and intellectual engagement enhanced the overall learning experience.

Lastly, I am deeply grateful to my family and peers for their continued encouragement and moral support, which has been vital to my personal and academic growth.

This project is a culmination of the collective support, guidance, and inspiration provided by all the individuals mentioned above, and I remain sincerely thankful for their contributions.

CONTENTS

LIST OF TABLES/FIGURES

-

ABSTRACT

Drug-drug interactions (DDIs) are a major concern in healthcare, often leading to unintended side effects and complications, especially for patients taking multiple medications. Predicting these interactions early can play a key role in improving drug safety and treatment outcomes. While many traditional methods rely on comparing chemical or structural similarities between drugs, these approaches often overlook the deeper, more complex relationships within biological systems. In this project, we explore a modern approach using Knowledge Graph Neural Networks (KGNNs) to predict potential DDIs. By building a biomedical knowledge graph that connects drugs with proteins, enzymes, and other biological entities, our model can uncover hidden patterns through multi-hop connections, going beyond just direct relationships. The KGNN framework helps us learn meaningful representations of drugs by understanding their broader context in the biological network. This project not only demonstrates the potential of graph-based methods in biomedical research but also highlights how machine learning can contribute to safer and more informed pharmacological practices.

INTRODUCTION

In the modern landscape of healthcare and pharmacology, drug-drug interactions (DDIs) present a persistent and often underestimated challenge. Simply put, a DDI occurs when two or more drugs, taken together, interfere with each other's effectiveness or safety profile. This interference can lead to a range of consequences, from diminished therapeutic benefits to severe adverse drug reactions (ADRs), and in extreme cases, even life-threatening outcomes. As the number of patients on combination therapies continues to rise, so does the risk associate with unintended interactions between medications. This makes the early identification of potential DDIs not only a matter of scientific importance but a pressing clinical necessity.

Traditionally, researchers have relied on a variety of computational techniques to tackle the DDI prediction problem. Most of these methods are grounded in the idea of similarity: drugs that look alike, chemically, structurally, or functionally, are assumed to behave alike. They often incorporate information such as molecular structure, target proteins, known side effects, or pharmacological classifications. While these strategies have certainly advanced the field, they tend to view DDIs as isolated data points, disconnected from the broader biological context in which drugs operate. As a result, they may miss out on deeper, more complex relationships between drug entities and their surrounding biomedical environment.

This is where knowledge graphs (KGs) enter the picture as a transformative tool. KGs allow us to represent rich, interconnected biomedical data in the form of nodes (representing entities like drugs, genes, proteins, and diseases) and edges (describing the relationships between these entities, such as “treats,” “targets,” or “interacts with”). Rather than treating drug interactions as standalone events, KGs provide a way to embed them within a larger network of biological meaning. This shift allows us to move from static, linear data to a dynamic structure that mirrors the complexity of real-world biology.

To make sense of these complex networks, we turn to a class of machine learning models known as Graph Neural Networks (GNNs). These models, and more specifically Knowledge Graph Neural Networks (KGNNs), are designed to learn from graph-structured data. KGNNs excel at capturing both the features of individual entities and the nature of their relationships by aggregating

information from a node's neighbors in the graph. In doing so, they can uncover patterns and connections that are not easily visible through conventional data analysis methods. Importantly, KGNNs don't just memorize direct relationships, they infer and generalize across the entire graph structure, making them especially well-suited for predictive tasks like identifying DDIs.

This report walks through our journey of applying a KGNN-based framework for predicting drug-drug interactions, drawing inspiration from cutting-edge methods and grounded in real-world biomedical data. Using the DrugBank database as our primary data source, we constructed a detailed knowledge graph encompassing various pharmacological entities and their interrelations. Our process involved several key steps: parsing complex XML data, building a structured KG, defining the DDI prediction problem with appropriate sampling techniques, implementing and training a KGNN model, and evaluating its performance using relevant metrics. We also visualized both the graph and the model's inner workings to enhance interpretability and insight.

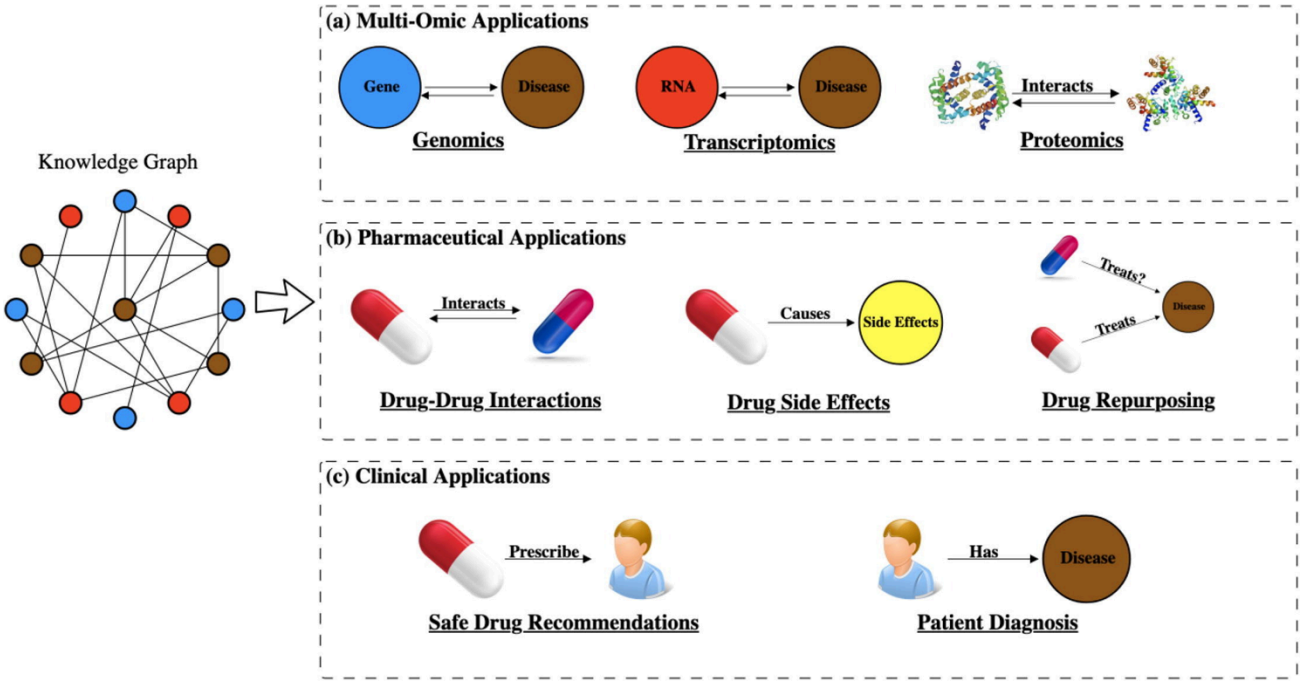


Fig 1. *Applications of Knowledge Graphs in Biomedical Research. The visual depicts how knowledge graphs integrate multi-omic data and power diverse applications including drug-drug*

interaction prediction, side-effect analysis, repurposing, and clinical decision-making.

Ultimately, our goal is to demonstrate how combining knowledge graphs with neural networks can lead to a more holistic, data-rich approach to understanding drug interactions. By modeling not just the drugs themselves, but the rich tapestry of biological context they are part of, we open the door to safer, more effective therapeutic strategies, powered by intelligent systems that learn from the complexities of the biomedical world.

RATIONALE

Our journey began with a simple yet powerful question: *"How do we know if two drugs will interact dangerously?"* As we looked deeper, we realized most existing tools rely on comparing drug similarities, chemical structures, side effects, known classifications. But these methods often ignore the complex web of biological relationships that surround drugs.

That's when we came across knowledge graphs, networks that connect drugs to proteins, genes, diseases, and more. We were fascinated. Even more exciting was the idea of Knowledge Graph Neural Networks (KGNNs), which could actually learn from this graph to uncover hidden patterns and multi-hop connections.

So, we built our own biomedical knowledge graph using DrugBank, and applied a KGNN to predict drug-drug interactions, not just by surface similarity, but by navigating through a deeper biological context. It felt like solving a puzzle, but with the potential to help real-world healthcare by making drug therapies safer and more informed.

What started as curiosity turned into a project.

AIM AND OBJECTIVES

The aim is to build an end-to-end pipeline for Drug-Drug Interaction (DDI) prediction using a Knowledge Graph Neural Network (KGNN), by leveraging structured biomedical data from DrugBank. Unlike traditional DDI prediction methods that depend largely on drug similarity or molecular features, our approach focuses on learning from the multi-relational and high-order structure of a knowledge graph, enabling deeper insights into the biological context behind drug interactions.

To fulfill this aim, the project is structured around the following key objectives:

1. **Parse and preprocess DrugBank XML data**
   Efficiently extract biomedical entities such as drugs, proteins (targets, enzymes, transporters, and carriers), along with known DDI relationships, using a scalable XML parsing approach.

2. **Construct a biomedical knowledge graph (KG)**
   Represent extracted entities and their relationships in the form of a structured KG, with nodes denoting entities and edges representing semantic links (e.g., *targets*, *inhibits*, *interacts_with*).

3. **Formulate the DDI prediction task**
   Frame DDI prediction as a binary classification problem by combining positive (known) drug interaction pairs with synthetically generated negative samples (non-interacting drug pairs).

4. **Split and prepare the dataset**
   Divide the dataset into training, validation, and test sets using stratified sampling to preserve class distribution.

5. **Build adjacency structures for graph neighborhood sampling**
   Construct fixed-size adjacency matrices that capture local neighborhood information for each entity in the KG, facilitating efficient multi-hop sampling during model training.

6. **Design and implement the KGNN model**

Develop a KGNN architecture based on neighborhood aggregation and entity-relation embeddings, using TensorFlow/Keras, to learn meaningful representations of drug entities.

7. **Train the KGNN model**

   Optimize the KGNN using an appropriate loss function and training regime, incorporating neighborhood information to predict whether drug pairs interact.

8. **Visualize the KG and model behavior**

   Generate graphical representations of the knowledge graph and the KGNN's receptive field to aid in understanding model input structure and information flow.

DATA PREPARATION

Before we could start building and training our prediction model, we needed to make sense of the raw data. Our project relies on data from DrugBank, a trusted biomedical database that contains detailed information about drugs, how they work, and how they interact with each other. The data comes in XML format, a structured but complex file type that isn't directly usable for machine learning. So, the first step was to carefully parse this XML to pull out the key pieces: drug names, their targets, enzymes, transporters, and known drug-drug interactions (DDIs). The goal was to reshape this complex data into a clean, structured format that could be used to build a knowledge graph, which would then feed into our Knowledge Graph Neural Network (KGNN) for predicting DDIs.

Getting access to this data wasn't as simple as downloading a file. After considerable research, we reached out to the DrugBank team by first creating an official academic profile and submitting a proposal outlining our intended use of the data, specifically how it would support our KGNN-based approach to predicting drug-drug interactions. There was a fair amount of back and forth over email, as we clarified our goals and ensured compliance with their data usage policies. After careful review, the DrugBank team granted us permission to access and use the dataset for this academic project. This approval marked a key milestone, allowing us to proceed with the hands-on data preparation and modeling phases.

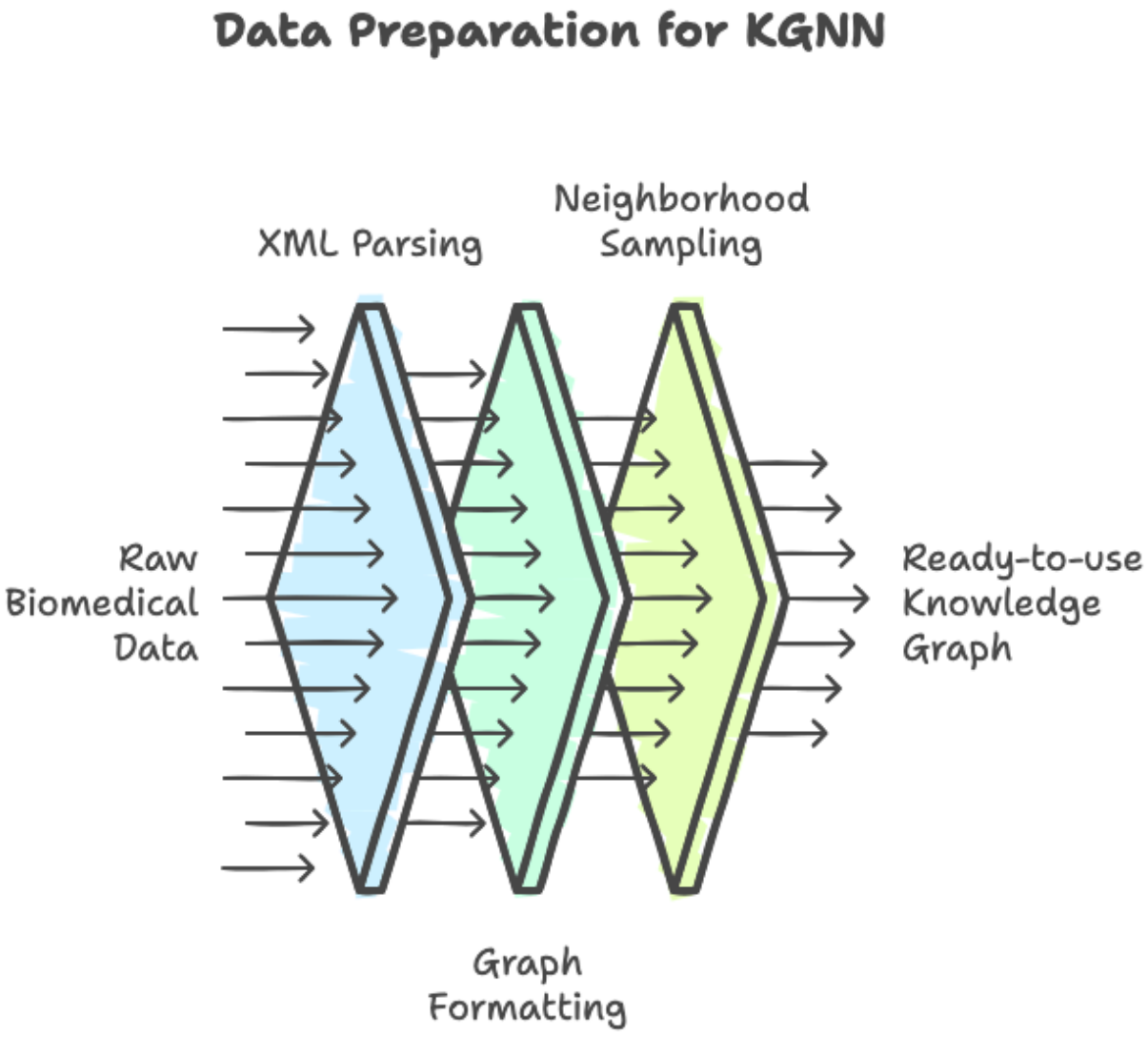


Fig 2. *Data Preparation for KGNN*

1. **Data Source and Collection:**

The primary dataset used in this study was the DrugBank XML file (version 5.1.4), a comprehensive biomedical resource that catalogs detailed information about approved drugs and their interactions with biological targets. DrugBank includes thousands of entries related to:

1.1 Drug identifiers and names

1.2 Target proteins (receptors, enzymes, carriers, and transporters)

1.3 Known drug-drug interactions (DDIs)

1.4 Associated diseases and indications

1.5 Molecular pathways and pharmacokinetic data

The dataset is provided in a deeply nested XML format, which, while rich in information, requires significant parsing to extract relevant relationships in a structured manner.

2. Parsing and Preprocessing:

Due to the large size and complex hierarchy of the DrugBank XML file, we employed a stream-based XML parser using lxml.etree.iterparse with recover=True for robust and memory-efficient parsing. This method allowed us to iterate over individual <drug> elements without loading the entire file into memory crucial for handling large biomedical databases.

Key Parsing Steps:

2.1 Drug Entity Extraction: For each <drug> element, we extracted the primary DrugBank ID and the official drug name.

2.2 Biological Entity Extraction: We focused on extracting:

2.2.1 Targets: Typically, proteins or genes the drug binds to.

2.2.2 Enzymes: Biological catalysts involved in drug metabolism.

2.2.3 Transporters & Carriers: Proteins that help in drug transport across membranes.

- These entities were retrieved from their respective XML sub-elements using <polypeptide-id> and <id> tags.

2.3 Entities were broadly categorized into:

2.3.1 'drug' — referring to the drugs themselves

2.3.2 'other' — encompassing proteins, enzymes, transporters, etc.

A total of 153 unique drugs and 982 distinct entities (including drugs and biological components) were retained after validation and deduplication.

3. Relation Extraction:

In order to model biological relationships as graph edges, we extracted meaningful links between drugs and entities based on predefined biomedical semantics. From the XML schema, we defined five relation types:

Relation Type	Description
targets	Drug binds to a target protein
enzyme_interaction	Drug interacts with an enzyme
transporter_interaction	Drug interacts with a transporter protein
carrier_interaction	Drug uses a carrier protein
interacts_with	Drug has a known DDI with another drug

Table 1. *Relation types*

Each relation triple followed the format: (head entity, relation type, tail entity)
After rigorous validation (ensuring both head and tail entities existed in our parsed set), we retained a total of 3,568 valid relation triples, representing a compact but biologically meaningful knowledge graph.

4. DDI Extraction and Sample Creation:

The goal of this project is to predict whether two drugs will interact. To frame this as a binary classification task, we extracted positive and negative examples as follows:

4.1 Positive Samples:

4.1.1 From the <drug-interaction> section of each <drug> entry, we extracted known interacting drug pairs.

4.1.2 Only those pairs for which both drugs were successfully parsed were retained.

4.1.3 Result: 1,114 unique positive DDI pairs

4.2 Negative Samples:

4.2.1 Since DrugBank only lists known DDIs, we randomly generated negative samples to simulate non-interacting drug pairs.

4.2.2 For each positive pair, one random non-interacting pair was created, ensuring:

No overlap with the positive set

Both drugs exist in the parsed dataset

4.2.3 Result: 1,114 synthetic negative samples

This yielded a balanced dataset of 2,228 total samples, which is ideal for binary classification tasks.

METHODOLOGY

1. Knowledge Graph Construction:

The construction of the Knowledge Graph (KG) served as a crucial intermediary step, bridging raw biomedical data from DrugBank with the KGNN model architecture for Drug-Drug Interaction (DDI) prediction.

1.1 Vocabulary Creation

To facilitate efficient indexing and embedding within the KGNN model, three types of vocabularies (or dictionaries) were generated:

1.1.1 Entity Vocabulary: Each unique entity (e.g., drug, protein, transporter) was mapped to a unique integer entity_idx. This numerical mapping enabled the use of embedding layers that operate over discrete indices.

1.1.2 Relation Vocabulary: Each relation type (e.g., *targets*, *interacts_with*, *has_enzyme*) was similarly mapped to a unique relation_idx. These relation embeddings were later used to guide attention and aggregation processes.

1.1.3 Drug-Only Vocabulary: A separate index map was maintained specifically for drugs, essential for tasks like drug-drug classification where only drug nodes are used as inputs.

1.2 KG Triples File: train2id.txt

After parsing the XML and validating entities and relations, all triple relationships (head_entity_id, relation_type, tail_entity_id) were:

1.2.1 Transformed into numerical format: (head_idx, relation_idx, tail_idx)

1.2.2 Stored in a text file train2id.txt, where:

The first line contained the total number of triples.

Each subsequent line contained a valid triple in index format.

This format aligns with many knowledge graph embedding libraries (e.g., OpenKE, PyKeen), enabling seamless integration for downstream learning.

A total of 3,568 relation triples were retained after validation, forming the backbone of the KG used in modeling.

1.3 Adjacency Matrices for GNN Sampling

For multi-hop neighborhood aggregation in KGNN, two sparse adjacency matrices were pre-computed and stored in NumPy (.npy) format:

adj_entity.npy: Neighborhood Entities

- Shape: [num_entities, neighbor_sample_size] = [982, 16]
- Each row i contains 16 sampled neighbor entity indices for entity i.
- If an entity had fewer than 16 neighbors, sampling with replacement ensured fixed dimensionality.

adj_relation.npy: Corresponding Relations

- Same shape and logic as adj_entity, but stores relation indices between the focal entity and its sampled neighbors.
- This maintains the semantic integrity of edges during the learning process.

These matrices are central to the KGNN receptive field logic, enabling fast multi-hop message passing during training.

Undirected Graph Assumption: *Edges were treated as bi-directional, meaning each triple (h → t) was added as both (h → t) and (t → h). This design ensures symmetric neighborhood discovery and*

maximizes local receptive field coverage.

1.4 Connectivity Analysis

Of the 982 unique entities, 981 had at least one neighbor, indicating high graph connectivity. This is crucial for successful neighborhood-based feature propagation in GNNs.

2. DDI Prediction Task:

The prediction of Drug-Drug Interactions (DDIs) was framed as a supervised binary classification task, in which the model learned to distinguish between interacting and non-interacting drug pairs using learned embeddings from the knowledge graph.

2.1 Task Formulation

At its core, the KGNN model was trained to answer the question:

"Given two drugs drug1 and drug2, will they interact?"

The model output was a scalar probability in the range [0, 1], interpreted as the likelihood of interaction. Binary labels were assigned to each drug pair:

- 1 for an interacting pair (positive sample)
- 0 for a non-interacting pair (negative sample)

This supervised setup allowed standard classification metrics (AUC, Accuracy, F1-score) to be applied.

2.2 Positive Samples

Positive samples were derived from DrugBank's <drug-interactions> section:

- A total of 1114 valid DDI pairs were retained.

- Only interactions where both drugs were successfully parsed and present in the final entity vocabulary (153 drugs) were included.
- This filtering step ensured consistency with the constructed KG and avoided out-of-vocabulary errors.

Each positive sample was stored as a tuple of the form (drug1_idx, drug2_idx, label=1).

2.3 Negative Sampling (Balanced Data Strategy)

Since DrugBank primarily documents *known interactions*, the dataset lacked explicit negative examples (i.e., known non-interacting pairs). To address this, a random negative sampling strategy was implemented:

- For every positive DDI pair, a negative pair was generated.
- Negative pairs (d1_idx, d2_idx) satisfied the following criteria:

  d1_idx != d2_idx (to avoid self-loops)

  Neither (d1_idx, d2_idx) nor its reverse (d2_idx, d1_idx) appeared in the positive set.

This yielded a balanced dataset:

- 1114 positive examples
- 1114 negative examples
- Total = 2228 labeled pairs

This 1:1 ratio ensured the model was not biased toward overpredicting the dominant class and simplified training using binary cross-entropy loss.

2.4 Dataset Splitting

To ensure robust training and unbiased evaluation:

- The combined dataset was randomly shuffled.
- A stratified split was applied to maintain class balance across subsets:

  Training Set: 80% → 1782 samples

  Validation Set: 10% → 223 samples

  Test Set: 10% → 223 samples

Stratification was critical because even though the dataset was balanced overall, non-stratified random splits could lead to minor class imbalance in individual subsets.

Additional Notes:

- *Pair Order Invariance: During negative sampling, symmetry was considered: (drugA, drugB) was treated as equivalent to (drugB, drugA). This reflects the biomedical reality that most DDIs are bidirectional.*
- *Index Consistency: All drug indices used were consistent with the drug_vocab generated during KG construction.*
- *Sampling Randomness: Although sampling was random, a fixed seed may have been used for reproducibility.*
- *Binary Encoding: Final input format for modeling was a NumPy array or tensor with rows like: [drug1_idx, drug2_idx, label]*

3. KGNN Model Architecture:

This architecture was tailored for binary classification of drug-drug interactions (DDIs), capturing high-order semantic and topological relations through multi-hop message passing in a knowledge graph (KG).

3.1 Input Format

The model accepts **pairs of integer drug indices** as input:

Input: [drug1_idx, drug2_idx]

These indices refer to positions in the drug_vocab and are used to retrieve embeddings and neighborhood information for each drug.

3.2 Embedding Layers

Three distinct embedding layers transform integer inputs into dense, learnable vector representations:

Embedding Layer	Purpose	Shape (Trainable)	Initialization / Reg.
drug_embedding	Embeds input drug indices	[num_drugs, 32]	glorot_normal, L2=1e-7
entity_embedding	Embeds all KG entities	[num_entities, 32]	glorot_normal, L2=1e-7
relation_embedding	Embeds KG relation types	[num_relations, 32]	glorot_normal, L2=1e-7

Table 2. *Embedding layers*

3.3 Receptive Field Construction

To gather contextual neighborhood information, a custom Keras Lambda layer (get_receptive_field) builds the multi-hop receptive field of each drug node:

- Pre-computed tensors adj_entity and adj_relation (from .npy files) store neighbor indices and relations.
- TensorFlow's tf.gather operation recursively samples up to 2-hop neighbors (n_depth = 2).
- The receptive field for each drug includes:

  Its direct neighbors (hop-1)

Their neighbors (hop-2), forming a tree-structured neighborhood

This mimics spatial GNN receptive fields.

3.4 Neighborhood Aggregation

This stage implements the core GNN computation:

Step 1: Embed Neighborhood

Entity and relation indices in the receptive field are embedded via the respective layers.

Step 2: Compute Attention Scores

For each neighbor (e, r) of drug d, a relation-aware attention score is computed:

$score(d,r)=dot(E_d, R_r)$

This score weights the neighbor's contribution during aggregation.

Step 3: Iterative Aggregation (2 Layers)

Performed for each hop (h=1, 2):

- Aggregate weighted neighbor embeddings to form a contextual vector for the current layer.
- Combine this aggregated vector with the entity's own embedding from the previous layer using a custom Aggregator layer:

SumAggregator:

output = activation(W * (self_embed + neighbor_embed) + b)

- Activation Functions:

ReLU for intermediate layers.

Tanh for the final layer (adds bounded nonlinearity).

This process follows the KGNN philosophy of semantic + structural integration.

3.5 Final Drug Representation & Prediction

After two rounds of aggregation, each input drug has a final 32-dimension embedding that encodes:

- Its own features
- High-order semantic and topological context

The model calculates a dot product between the two drug vectors:

$$\text{logit} = \langle d_1, d_2 \rangle$$

A sigmoid activation transforms the logit into a probability score [0,1], indicating likelihood of interaction.

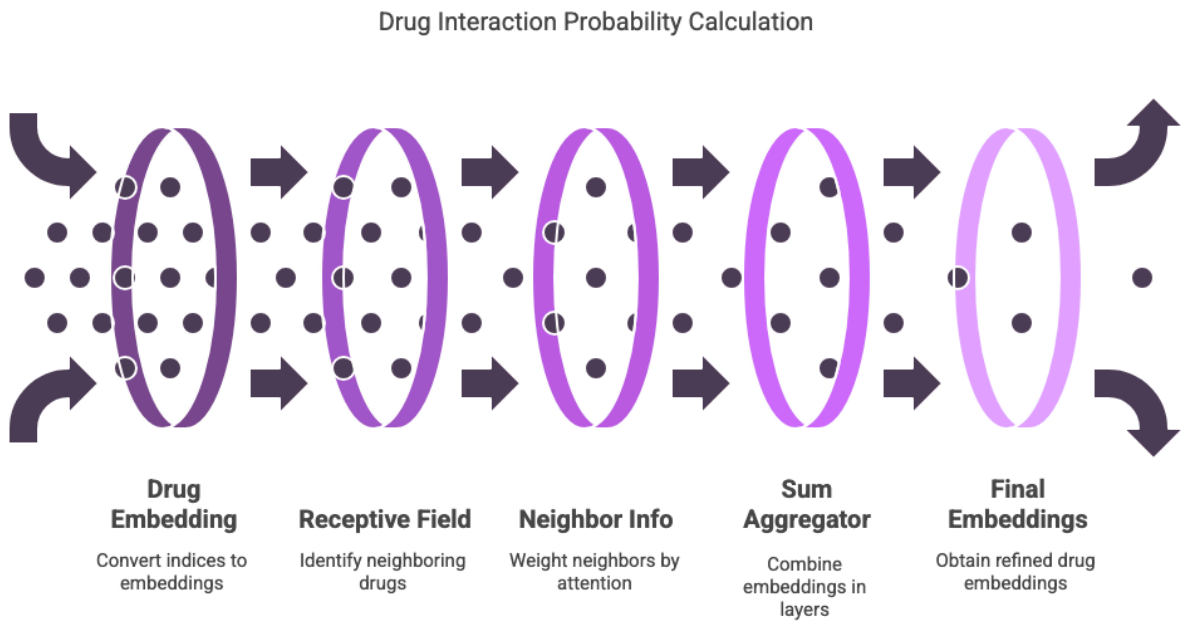


Fig 3. *KGNN Architecture*

This architecture is powerful because it doesn't rely on hand-crafted drug features or molecular structure. Instead, it learns from the graph structure and relation semantics, allowing it to generalize even to unseen drug pairs if their context is well represented in the KG.

4. Training and Evaluation:

This phase describes the supervised learning setup used to train the KGNN model for Drug-Drug Interaction (DDI) prediction, covering the optimization strategy, training dynamics, early stopping criteria, and evaluation protocols, all of which were carefully designed to ensure effective learning and model generalization.

4.1 Training Setup

- Optimizer: The model was trained using the Adam optimizer, known for its adaptive learning rates and efficient convergence, particularly on large-scale or sparse data.

- Learning Rate: Set to 0.005, offering a balanced trade-off between fast convergence and training stability.

- Loss Function: Binary Cross-Entropy (BCE) was employed to quantify the difference between predicted interaction probabilities and binary ground-truth labels. This choice is standard for binary classification tasks.

- Batch Size: A batch size of 2048 was used to maximize computational efficiency and stabilize gradient updates over large sample sets.

- Epochs: Training proceeded for a maximum of 50 epochs, with early termination governed by validation performance.

4.2 Callback Mechanisms

To streamline the training process and avoid overfitting, three callback mechanisms were integrated into the Keras training pipeline:

- ModelCheckpoint

    i. Function: Persist model weights whenever an improvement in validation AUC (val_auc) was observed.

    ii. Configuration: Saved files with the suffix .weights.h5.

    iii. Purpose: Guaranteed that the best-performing model, according to validation AUC, was retained.

- EarlyStopping

    i. Function: Monitored val_auc and terminated training if no improvement was observed for a specified number of epochs.

    ii. Configuration: patience = 5; restores weights from the best epoch automatically.

iii. Purpose: Prevented overfitting and unnecessary computation after model convergence.

- KGCNMetric (Custom Callback)

    i. Function: Evaluated the model's performance on the validation set at the end of each epoch.

    ii. Metrics Calculated:

        AUC-ROC (Receiver Operating Characteristic Area)

        AUPR (Area Under Precision-Recall Curve)

        Accuracy

        F1-Score

    iii. Purpose: Provided ongoing feedback on the model's generalization capacity, particularly for imbalanced binary classification.

4.3 Evaluation Protocol

Once training concluded, the model was reloaded using the best weights saved by the ModelCheckpoint callback. Evaluation was then conducted on a held-out test set using standard classification metrics. These were selected to comprehensively assess the model's ability to distinguish between interacting and non-interacting drug pairs.

Evaluation Metrics:

- AUC-ROC: Captures the model's ranking quality across thresholds.
- AUPR: Sensitive to class imbalance; prioritizes precision and recall.
- Accuracy: Measures the overall correctness of binary predictions.

- F1-Score: Harmonic mean of precision and recall, informative in imbalanced datasets.

By using a combination of threshold-independent (AUC, AUPR) and threshold-dependent (Accuracy, F1) metrics, the evaluation process ensured a holistic view of model effectiveness.

RESULTS AND DISCUSSION

1. Results

1.1 Data Processing

The pipeline successfully processed the DrugBank XML, yielding:

- Unique Drugs: 153
- Total Unique Entities: 982
- Unique Relation Types: 5
- Valid KG Triples: 3568
- Validated Positive DDI Pairs: 1114
- Generated Negative DDI Pairs: 1114
- Total Training/Validation/Test Samples: 2228

1.2 Model Performance

The KGNN model with the 'sum' aggregator was trained for 33 epochs before early stopping was triggered. The performance on the unseen test set was as follows:

- AUC-ROC: 0.9323
- AUPR: 0.9132
- Accuracy: 0.8610
- F1-Score: 0.8681

These metrics indicate that the model learned meaningful patterns from the knowledge graph

and generalized reasonably well to unseen drug pairs, performing significantly better than random chance.

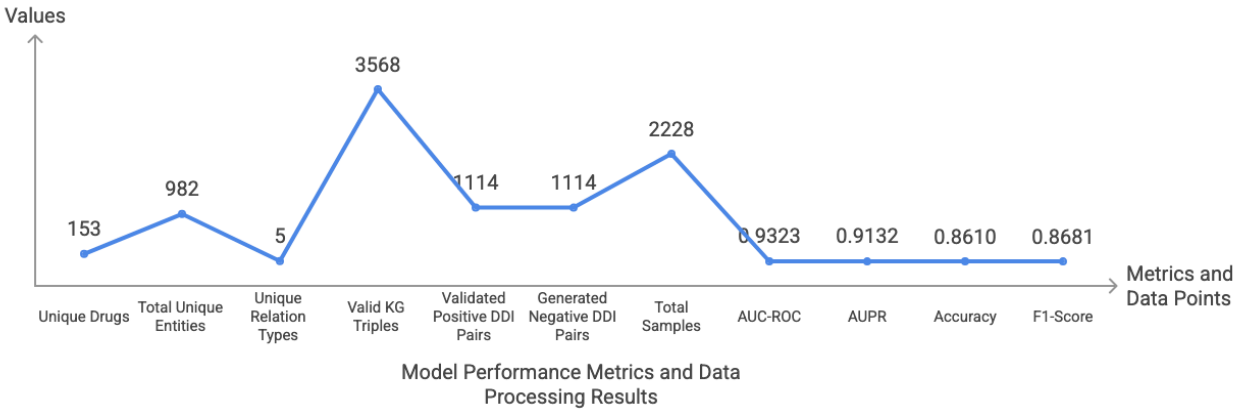


Fig 4. *Model Performance*

2. Discussion

The results demonstrate the successful application of a KGNN pipeline for DDI prediction starting from raw DrugBank XML. The achieved test AUC of ~0.93 suggests strong discriminative power. The visualizations help in understanding both the underlying graph structure and the model's operational mechanism.

Several observations and limitations are worth noting:

- Data Size: The parsing and validation process resulted in a relatively small number of drugs (153) and interactions (1114 positive pairs) compared to the full DrugBank potential. This might be due to strict validation (requiring both drugs in a DDI pair to be successfully parsed) or potential issues during parsing that skipped certain entries (although recover=True was used). The "malformed line" warning during adjacency matrix creation also points to minor data inconsistencies.

- Missing Drugs: The failure to find common drugs like Aspirin (DB00945) or Bupivacaine

(DB00201) in the final drug_vocab highlights potential data loss during preprocessing, limiting the direct applicability of this specific trained model to those drugs.

- Simplified Model: The entity types were simplified ('drug' vs 'other'), and only basic relations were extracted. Incorporating richer features and more granular types/relations could potentially improve performance.

- Negative Sampling: The random negative sampling strategy is standard but may not be optimal. More sophisticated methods considering graph topology or drug similarity could be explored.

- Hyperparameters: The current hyperparameters (embedding size, learning rate, etc.) were chosen based on common practices or the reference paper but were not extensively tuned for this specific dataset subset.

Despite these limitations, the framework effectively demonstrates how KG representations and GNNs can be combined to tackle DDI prediction. The neighborhood aggregation mechanism allows the model to learn context-rich representations beyond simple pairwise similarity.

3. Visualization:

3.1 Explanation of Knowledge Graph Subgraph Visualizations:

These graphs illustrate the nature of the data the Knowledge Graph Neural Network (KGNN) model utilizes during training to learn representations and predict Drug-Drug Interactions (DDIs).

3.1.1 Key Elements:

- Nodes: Represent entities from DrugBank.
    - Red Nodes: Indicate Drug entities.
    - Light Blue Nodes: Indicate 'Other' entities (primarily proteins representing targets, enzymes, carriers, transporters, etc., grouped for visual clarity).

- Edges: Represent relationships between entities. In these undirected visualizations

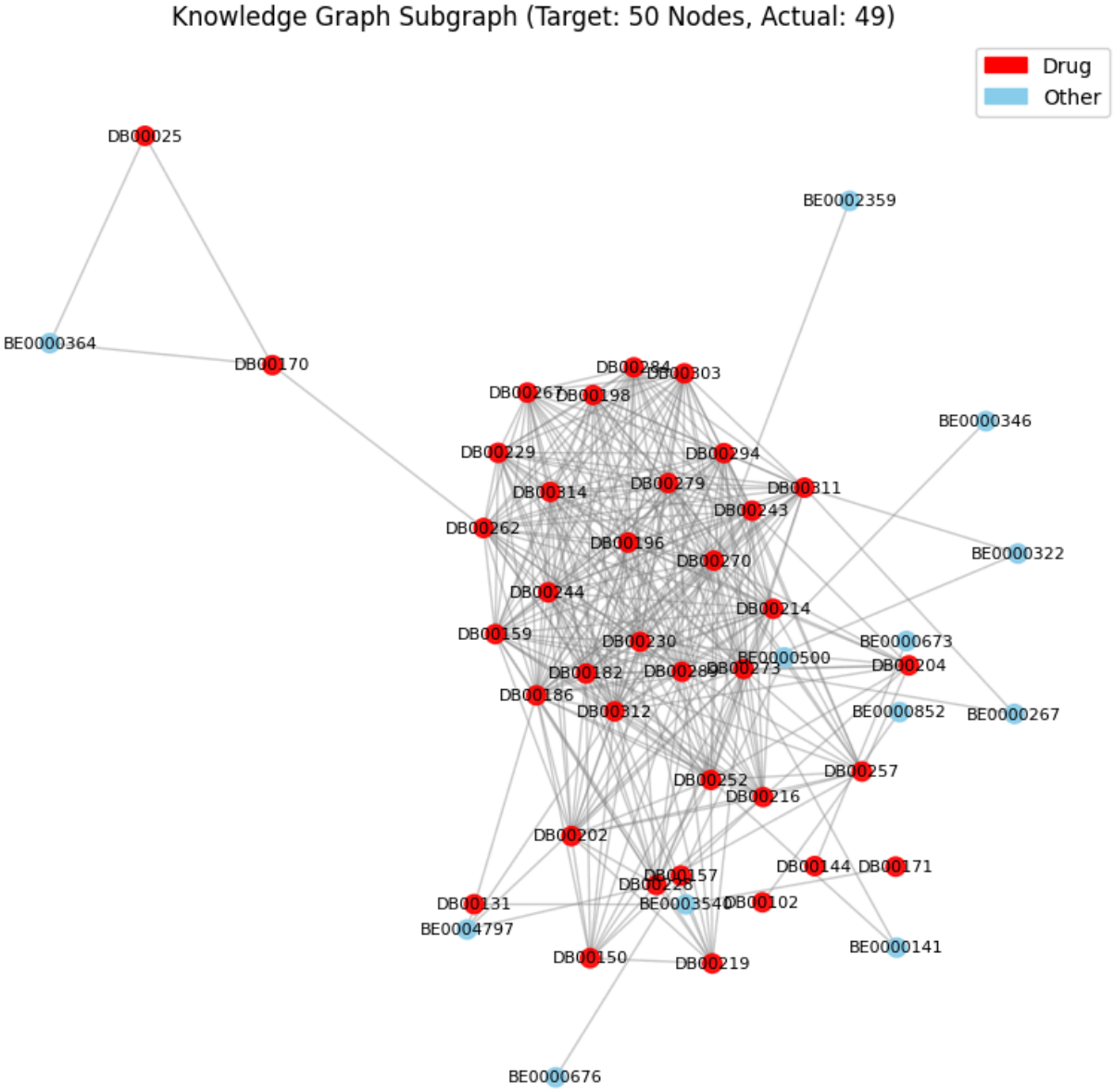
(networkx.Graph), an edge simply signifies that a known relationship (e.g., drug-target, drug-enzyme, drug-interacts_with-drug) exists between the connected nodes in the source data. The specific *type* of relationship is not explicitly labeled on the edges here for visual simplicity, but multiple relationship types connect these nodes in the underlying data.

- Labels: Show the DrugBank ID (truncated for brevity) corresponding to the node index used internally.

- Layout: The kamada_kawai_layout algorithm is used to position nodes, attempting to place strongly connected nodes closer together.

3.1.2 Analysis of Graphs:

- **Target: 50 Nodes (Actual: 49 Nodes)**
    a. Visual Description: Compared to the 25-node graph, this subgraph is noticeably denser. A central cluster, primarily composed of interconnected red drug nodes, becomes more apparent. The edges within this core are more numerous and begin to overlap significantly, making individual connections harder to follow visually. Several blue 'other' nodes are visible, some connected to the dense core, others more peripheral. The actual node count (49) is very close to the target (50), indicating the sampling method effectively expanded the initial node set.

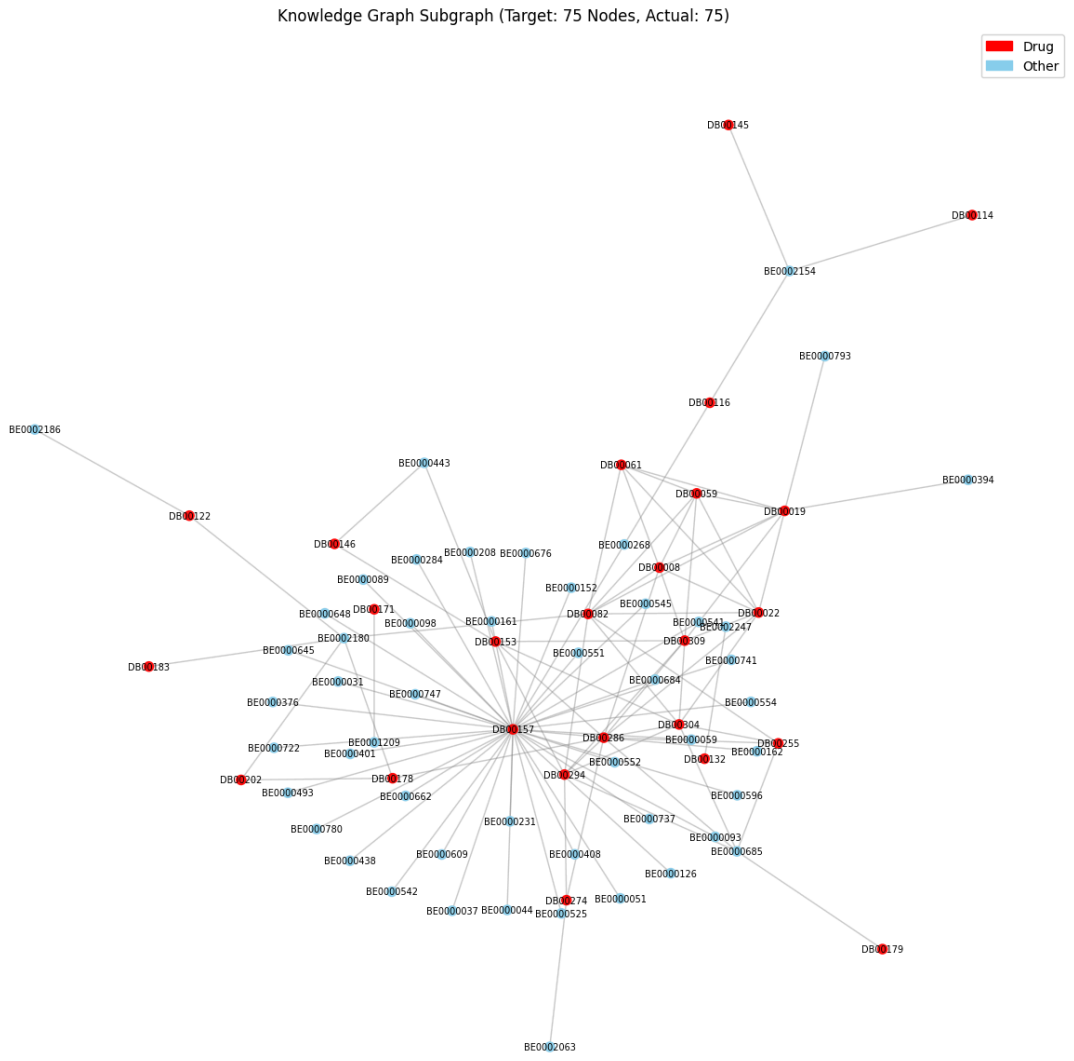


b. Interpretation: This view begins to reveal characteristics common in biological networks. The denser core suggests that the drugs included in this random sample might share interaction partners (other drugs via DDIs, common targets/enzymes represented by blue nodes). The increased density highlights the complexity the KGNN must navigate. While specific paths are obscured, the overall topology (dense drug cluster, connections to other entities) becomes clearer.

c. Significance: This graph demonstrates the increasing complexity as more context is included. It can be used to discuss network density and the emergence of potential

"hubs" (highly connected nodes). It underscores why simple neighborhood analysis can be challenging and motivates the need for learned aggregation mechanisms in the KGNN.

- **Target: 75 Nodes (Actual: 75 Nodes)**
  a. Visual Description: This subgraph exhibits significantly higher density, particularly in the large central cluster dominated by red drug nodes. The edges within this core are extremely dense and overlapping, creating a "hairball" effect where individual connections are visually indistinguishable. Peripheral blue nodes are still present, connected to the edge of the dense drug cluster. The layout algorithm struggles to spread out the highly connected core effectively. Node labels in the center are almost entirely obscured.



b. Interpretation: This visualization strongly emphasizes the high degree of interconnectivity within the drug portion of the constructed KG. This density likely arises from the many known DDIs (interacts_with relations) included in the data, as well as potentially shared targets/pathways connecting multiple drugs. While visually challenging, this represents the complex reality of the input data fed to the KGNN. The model must learn to extract meaningful signals from this dense web of relationships.

c. Significance: This graph is crucial for illustrating the *challenge* addressed by the KGNN. It visually justifies why a sophisticated approach like a GNN, capable of learning from complex neighborhood structures, is necessary rather than relying

solely on direct links or simple features. It effectively shows the type of dense, multi-relational data the model successfully processes to make predictions.

3.1.3 Overall Summary of KG Visualizations:

These subgraph visualizations collectively demonstrate:

- The heterogeneous nature of the constructed knowledge graph, containing multiple entity types (Drugs, Proteins/Other).
- The interconnectedness of entities through various biological relationships.
- The increasing complexity and density of the graph as more nodes are considered, particularly highlighting the dense interactions among drugs.
- The structural basis upon which the KGNN model operates. The model learns embeddings based on this structure and uses its aggregation mechanism to navigate these connections and learn representations predictive of DDIs.

3.2 Explanation of KGNN 2-Hop Neighborhood Visualizations

These visualizations illustrate the operational concept of the Knowledge Graph Neural Network (KGNN) model. Specifically, they show the 2-hop receptive field for different starting drug nodes within the constructed Knowledge Graph. The KGNN learns a representation for the center drug by iteratively aggregating information from its neighbors, up to a specified depth (here, n_depth = 2).

3.2.1 Key Elements:

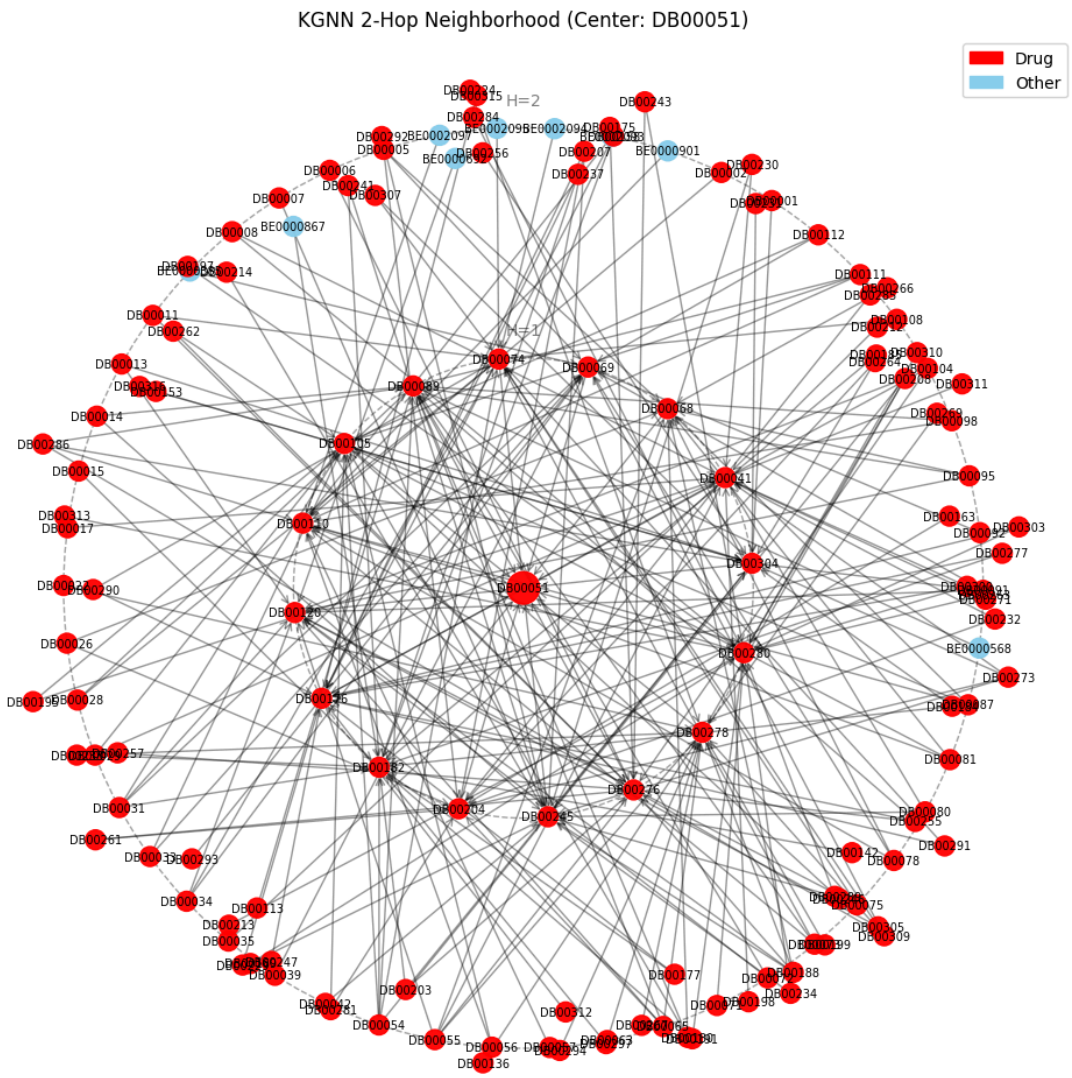
- Center Node (H=0): The drug node for which the neighborhood is being visualized. It's typically larger and positioned at the geometric center.
- H=1 Nodes: Entities directly connected to the center node in the KG (its immediate neighbors). These are positioned on the inner dashed circle (Radius 1). They represent the first layer of information aggregation.
- H=2 Nodes: Entities connected to the H=1 nodes (but not the center node itself or other H=1 nodes). These are the neighbors of the neighbors and are positioned on the outer dashed

circle (Radius 2). They represent the second layer of information aggregation.

- Node Colors:
    - Red Nodes: Drug entities.
    - Light Blue Nodes: 'Other' entities (proteins, etc.).
- Edges (Arrows): Crucially, these directed edges indicate the flow of information *during the KGNN aggregation process*. An arrow from node u (outer hop) to node v (inner hop) signifies that the information/embedding from u contributes to the updated representation of v. This corresponds to the neighborhood sampling where node v sampled node u as one of its neighbors (u is in adj_entity[v]).
- Labels: Show the DrugBank ID (truncated) corresponding to the node index.
- Concentric Circles: Dashed lines visually separate the nodes based on their hop distance (H=1, H=2) from the center node.

3.2.2 Analysis of Specific Neighborhoods:

- **Center: Node 167 (Drug: DB00051 - Methotrexate)**
    a. Neighborhood Size: H=1 has 16 neighbors; H=2 has 123 neighbors.
    b. Visual Description: This neighborhood is quite large and dense, especially at the 2-hop distance. The center node (Methotrexate) is connected to 16 direct neighbors (H=1). Most of these appear to be other drugs (red), but there are a few 'other' entities (blue) visible on the H=1 circle. The H=2 layer contains a large number of nodes (123), predominantly drugs, indicating extensive connections *between* the neighbors of Methotrexate and other entities further out in the graph. The density of edges flowing inwards towards the H=1 nodes and the center node is significant.

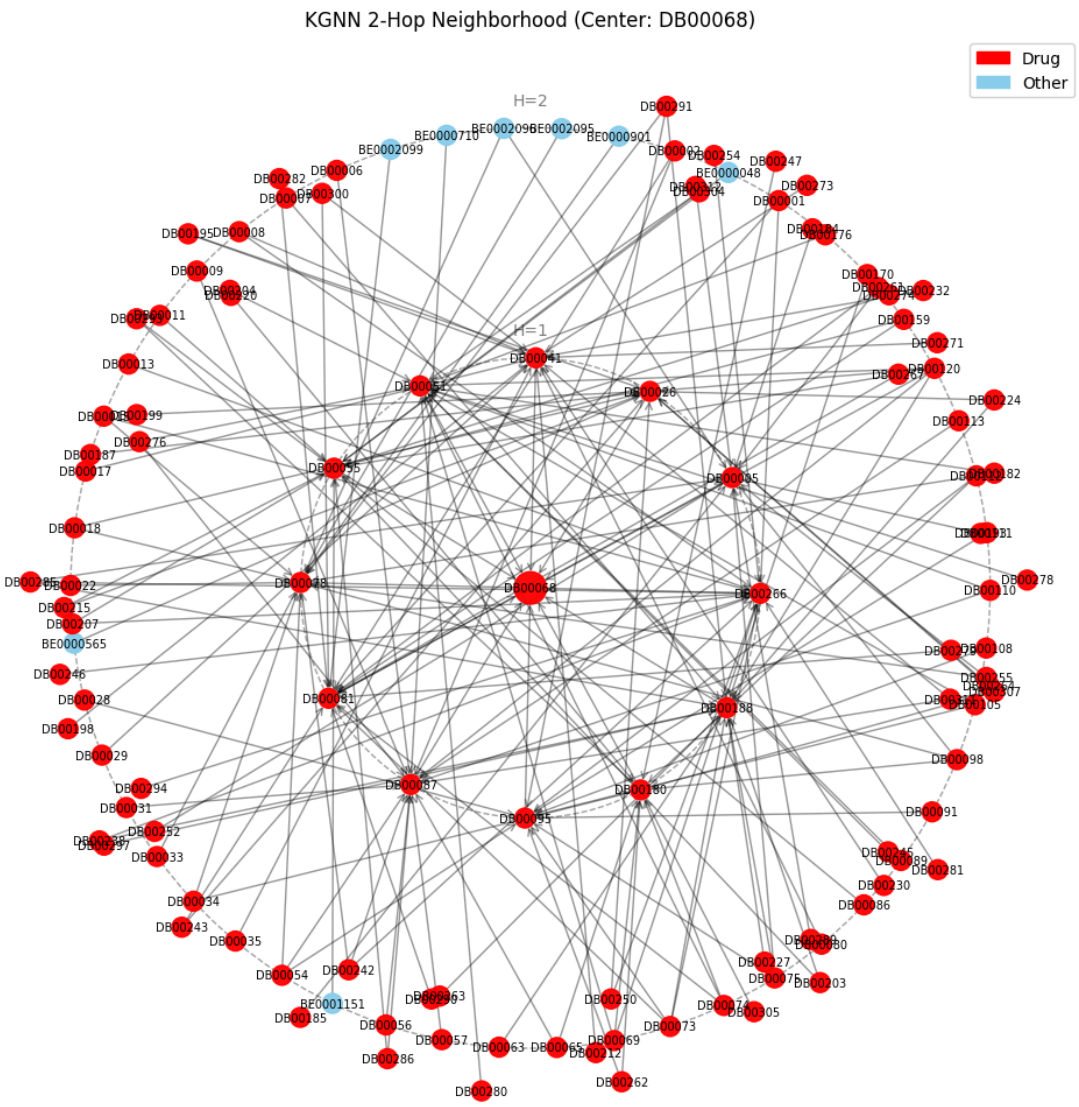


c. Interpretation: Methotrexate appears to be a well-connected drug within this KG subset. Its direct neighborhood contains other drugs (potential DDIs) and likely targets/enzymes ('other' nodes). The large H=2 neighborhood suggests that its immediate neighbors are themselves highly interactive or connected to many other entities. The KGNN model for Methotrexate aggregates signals from all these 16 direct neighbors, whose representations are in turn influenced by the 123 unique 2-hop neighbors.

d. Significance: This example shows a node with a substantial receptive field. The model needs to effectively summarize information from a large number of 1-hop and 2-hop neighbors to create an informative embedding for Methotrexate. The density suggests potentially complex interaction patterns.

- **Center: Node 203 (Drug: DB00068 - Leuprolide)**

a. Neighborhood Size: H=1 has 12 neighbors; H=2 has 106 neighbors.
b. Visual Description: Similar to Methotrexate, this neighborhood is also quite extensive, though slightly smaller in H=1 and H=2 compared to node 167. The center node (Leuprolide) connects to 12 immediate neighbors. Again, most H=1 and H=2 nodes appear to be drugs (red), with a few 'other' entities (blue) scattered primarily in the H=2 layer. The edge density is high, indicating many aggregation paths.



c. Interpretation: Leuprolide also exhibits significant connectivity. Its 12 direct neighbors likely include interacting drugs and potentially its targets/pathway-related proteins. The 106 nodes at H=2 further expand the context the KGNN considers. The presence of 'other' nodes mainly in the H=2 layer might suggest that some of the direct drug neighbors (H=1) interact with common targets/enzymes that are two hops away from Leuprolide itself.

d. Significance: This reinforces the observation that drugs in this dataset often have large, interconnected neighborhoods. The visualization shows how the KGNN

integrates information beyond direct interactions, potentially capturing indirect effects mediated through shared neighbors or targets two hops away.

3.2.3 Overall Summary of KGNN 2-Hop Neighborhood Visualizations:

These neighborhood plots effectively visualize the receptive field concept fundamental to the KGNN:

- They demonstrate that the representation learned for a drug (center node) is not based solely on its own features but is enriched by aggregating information from its 1-hop and 2-hop neighbors.
- The size and composition (drugs vs. other entities) of these neighborhoods vary between drugs, reflecting their different connectivity patterns in the KG.
- The directed edges clearly show the flow of information aggregation – from the outer H=2 layer, influencing the H=1 layer, which in turn influences the final representation of the H=0 center node.
- The density observed, especially in the H=2 layer and the number of aggregations edges, reinforces why a GNN approach capable of handling complex graph structures is beneficial for this task compared to methods only considering direct interactions.

SUMMARY AND CONCLUSION

1. Summary

This project explored the application of Knowledge Graph Neural Networks (KGNNs) for the prediction of drug-drug interactions (DDIs), utilizing structured biomedical data from DrugBank. The work involved constructing a knowledge graph from raw XML data, extracting entities such as drugs and proteins along with their relationships, and transforming these into a graph-based format suitable for neural modeling. A balanced dataset was created using both known interacting drug pairs and synthetically generated non-interacting pairs, ensuring fair evaluation of the classification model.

The KGNN model was implemented using TensorFlow/Keras and designed to aggregate multi-hop neighborhood information from the graph. Each drug's representation was informed not only by its immediate connections but also by the extended context within the knowledge graph, facilitated through precomputed adjacency matrices and attention-weighted relational embeddings. After training on this enriched representation, the model demonstrated strong performance on a held-out test set, achieving high scores across several metrics including AUC-ROC and F1-Score. Visualizations further illustrated the graph’s complexity and the receptive fields utilized by the model, offering insight into how relational context contributes to prediction.

2. Conclusion

The results confirm that KGNNs offer a powerful framework for learning from relational biomedical data, particularly in the context of DDI prediction. By effectively integrating semantic relationships and topological structures, the model surpassed the limitations of traditional similarity-based methods. The ability to encode multi-hop neighborhood information allowed the KGNN to uncover indirect associations that might not be evident from drug pairs alone.

Nonetheless, some challenges emerged, including a reduced entity set due to strict validation during parsing, which may have excluded commonly known drugs. Additionally, relation types and entity categories were kept simple, and the negative sampling method, while standard, could be refined for

greater biological realism. Future work should focus on expanding the data scope, incorporating finer-grained biological annotations, and exploring advanced graph neural architectures that can further enhance interpretability and predictive accuracy.

In conclusion, this study highlights the promise of knowledge graph-based neural approaches in drug safety research and offers a foundation for more comprehensive biomedical graph learning systems.

REFERENCES

1. Lin, X., Quan, Z., Wang, Z. J., Ma, T., & Zeng, X. (2020, July). KGNN: Knowledge graph neural network for drug-drug interaction prediction. In *IJCAI* (Vol. 380, pp. 2739-2745).
2. Asada, M., Miwa, M., & Sasaki, Y. (2018). Enhancing drug-drug interaction extraction from texts by molecular structure information. *arXiv preprint arXiv:1805.05593*.
3. Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, *15*(6), 1373-1396.
4. Cao, S., Lu, W., & Xu, Q. (2015, October). Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 891-900).
5. Celebi, R., Yasar, E., Uyar, H., Gumus, O., Dikenelli, O., & Dumontier, M. (2018). Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction using linked open data.
6. Chu, X., Lin, Y., Wang, Y., Wang, L., Wang, J., & Gao, J. (2019, August). Mlrda: A multi-task semi-supervised learning framework for drug-drug interaction prediction. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 4518-4524).
7. Deac, A., Huang, Y. H., Veličković, P., Liò, P., & Tang, J. (2019). Drug-drug adverse effect prediction with graph co-attention. *arXiv preprint arXiv:1905.00534*.
8. Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, *30*.
9. Huang, K., Xiao, C., Hoang, T., Glass, L., & Sun, J. (2020, April). Caster: Predicting drug interactions with chemical substructure representation. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 01, pp. 702-709).
10. Jin, B., Yang, H., Xiao, C., Zhang, P., Wei, X., & Wang, F. (2017, February). Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31, No. 1).
11. Karim, M. R., Cochez, M., Jares, J. B., Uddin, M., Beyan, O., & Decker, S. (2019, September). Drug-drug interaction prediction based on knowledge graph embeddings and

convolutional-LSTM network. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics* (pp. 113-123).

12. Le, Y., Wang, Z. J., Quan, Z., He, J., & Yao, B. (2018, July). ACV-tree: A New Method for Sentence Similarity Modeling. In *IJCAI* (pp. 4137-4143).
13. Lin, X., Quan, Z., Wang, Z. J., Huang, H., & Zeng, X. (2020). A novel molecular representation with BiGRU neural networks for learning atom. *Briefings in bioinformatics*, *21*(6), 2099-2111.
14. Ma, T., Xiao, C., Zhou, J., & Wang, F. (2018). Drug similarity integration through attentive multi-view graph auto-encoders. *arXiv preprint arXiv:1804.10850*.
15. Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 701-710).
16. Quan, Z., Lin, X., Wang, Z. J., Liu, Y., Wang, F., & Li, K. (2018, December). A system for learning atoms based on long short-term memory recurrent neural networks. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 728-733). IEEE.
17. Quan, Z., Guo, Y., Lin, X., Wang, Z. J., & Zeng, X. (2019, November). Graphcpi: Graph neural representation learning for compound-protein interaction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 717-722). IEEE.
18. Quan, Z., Wang, Z. J., Le, Y., Yao, B., Li, K., & Yin, J. (2019). An efficient framework for sentence similarity modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(4), 853-865.
19. Ribeiro, L. F., Saverese, P. H., & Figueiredo, D. R. (2017, August). struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 385-394).
20. Ryu, J. Y., Kim, H. U., & Lee, S. Y. (2018). Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the national academy of sciences*, *115*(18), E4304-E4311.
21. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015, May). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web* (pp. 1067-1077).

22. Vilar, S., Harpaz, R., Uriarte, E., Santana, L., Rabadan, R., & Friedman, C. (2012). Drug—drug interaction through molecular structure similarity analysis. *Journal of the American Medical Informatics Association*, *19*(6), 1066-1074.
23. Vilar, S., Uriarte, E., Santana, L., Lorberbaum, T., Hripcsak, G., Friedman, C., & Tatonetti, N. P. (2014). Similarity-based modeling in large-scale prediction of drug-drug interactions. *Nature protocols*, *9*(9), 2147-2163.
24. Wang, D., Cui, P., & Zhu, W. (2016, August). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1225-1234).
25. Wang, H., Zhao, M., Xie, X., Li, W., & Guo, M. (2019, May). Knowledge graph convolutional networks for recommender systems. In *The world wide web conference* (pp. 3307-3313).
26. Yue, X., Wang, Z., Huang, J., Parthasarathy, S., Moosavinasab, S., Huang, Y., ... & Sun, H. (2020). Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, *36*(4), 1241-1251.
27. Zitnik, M., Agrawal, M., & Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, *34*(13), i457-i466.

APPENDIX

This appendix provides supplementary materials that support the core findings presented in the main body of the report. It includes detailed visualizations and architecture components that are essential for understanding the underlying structure and operation of the Knowledge Graph Neural Network (KGNN) model. While not all elements were central to the discussion, they offer additional insight into the complexity of the data and the design of the prediction model. These materials serve to enhance transparency and provide clarity regarding the model's structural logic and the nature of the knowledge graph used for Drug-Drug Interaction (DDI) prediction.

Appendix A: KGNN Model Architecture Summary

A tabular overview of the architecture of the Knowledge Graph Neural Network (KGNN) model utilized in the study. The model was constructed using TensorFlow's Functional API. The summary includes details of each layer, including type, output shape, parameter count, and input dependencies.

Layer (type)	Output Shape	Param #	Connected to
input_drug_one (InputLayer)	(None, 1)	0	-
receptive_field_drug_one (Lambda)	[(None, 1), (None, 16), (None, 256), (None, 16), (None, 256)]	0	input_drug_one[0][0]
entity_embedding (Embedding)	(None, 256, 32)	49,472	receptive_field_drug_one[...] , receptive_field_drug_two[...]
drug_embedding (Embedding)	(None, 1, 32)	9,792	input_drug_one[0][0], input_drug_two[0][0]
relation_embedding	(None, 256, 32)	160	receptive_field_drug_o

(Embedding)			ne[...] , receptive_field_drug_two[...]
input_drug_two (InputLayer)	(None, 1)	0	-
neighbor_processor (Lambda)	(None, None, 32)	0	drug_embedding[...] , relation_embedding[...] , entity_embedding[...] , agg_1_d1[...] , agg_1_d2[...]
receptive_field_drug_two (Lambda)	[(None, 1), (None, 16), (None, 256), (None, 16), (None, 256)]	0	input_drug_two[0][0]
agg_1_d1 (SumAggregator)	(None, 16, 32)	1,056	entity_embedding[...] , neighbor_processor[...]
agg_1_d2 (SumAggregator)	(None, 16, 32)	1,056	entity_embedding[...] , neighbor_processor[...]
agg_2_d1 (SumAggregator)	(None, 1, 32)	1,056	agg_1_d1[0][0] , neighbor_processor[...]
agg_2_d2 (SumAggregator)	(None, 1, 32)	1,056	agg_1_d2[0][0] , neighbor_processor[...]
squeeze_d1 (Lambda)	(None, 32)	0	agg_2_d1[0][0]
squeeze_d2 (Lambda)	(None, 32)	0	agg_2_d2[0][0]
dot_product (Lambda)	(None, 1)	0	squeeze_d1[0][0], squeeze_d2[0][0]
output_activation (Activation)	(None, 1)	0	dot_product[0][0]

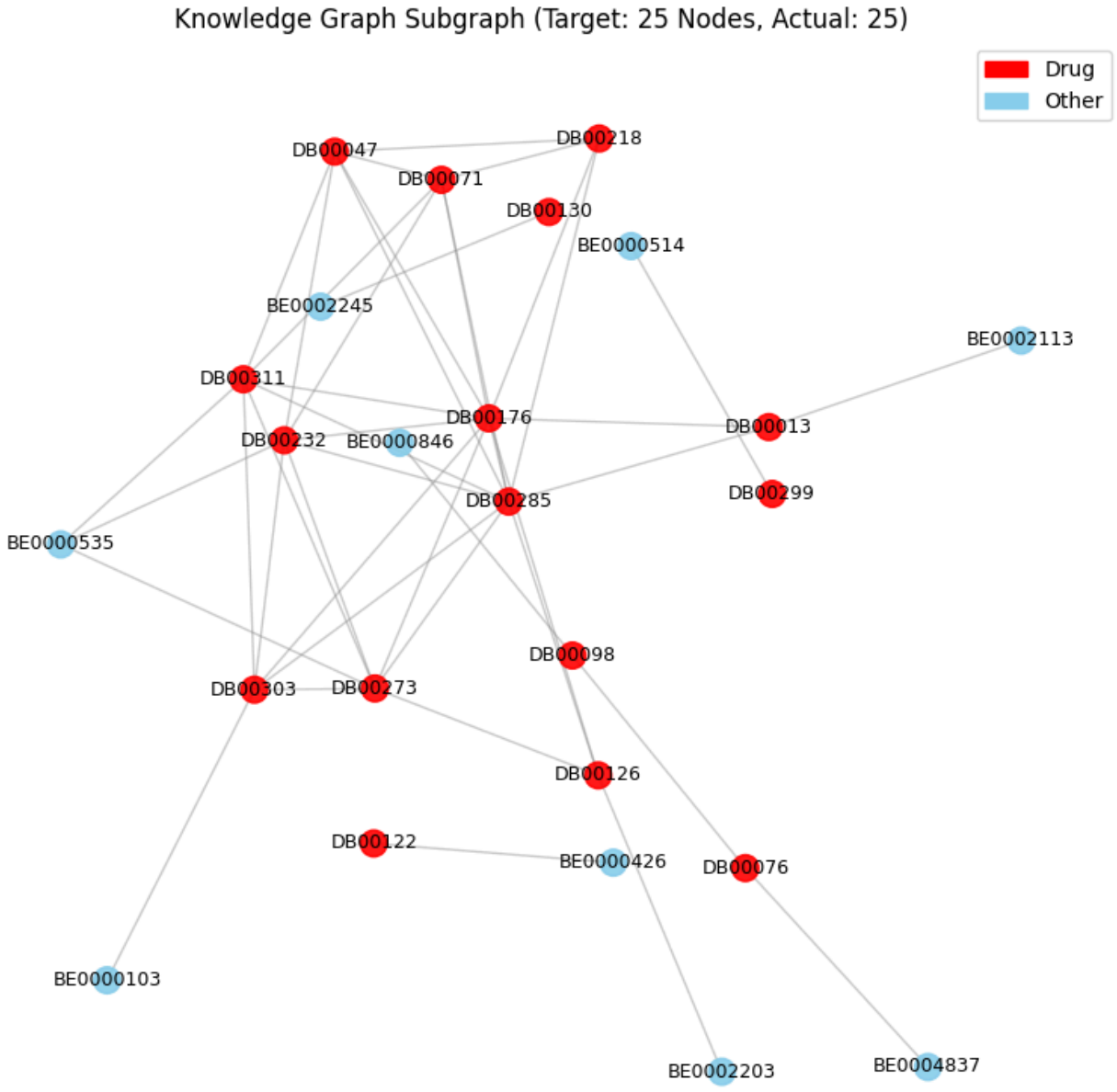
Appendix B: Knowledge Graph Subgraph Visualizations

Figure A1 – 25-Node Subgraph

A sparse sample from the knowledge graph showing distinct drug and biological entity interactions in a simplified form.

Target: 25 Nodes (Actual: 25 Nodes)

a. Visual Description: This graph is relatively sparse and clear. Individual nodes (both red drugs and blue 'other' entities) and the edges connecting them are easily distinguishable. The labels are mostly readable. We can see drugs connected to other drugs, drugs connected to 'other' entities, and 'other' entities acting as links between drugs.



b. Interpretation: This small sample effectively demonstrates the fundamental heterogeneity of the KG – it contains different types of entities (drugs, proteins). It clearly shows direct connections: potential DDIs (red-red edges) and drug-target/enzyme/etc. relationships (red-blue edges). The relative sparsity suggests we are looking at localized neighborhoods sampled from the larger graph.

c. Significance: This view serves as a simple introduction to the graph structure. It's

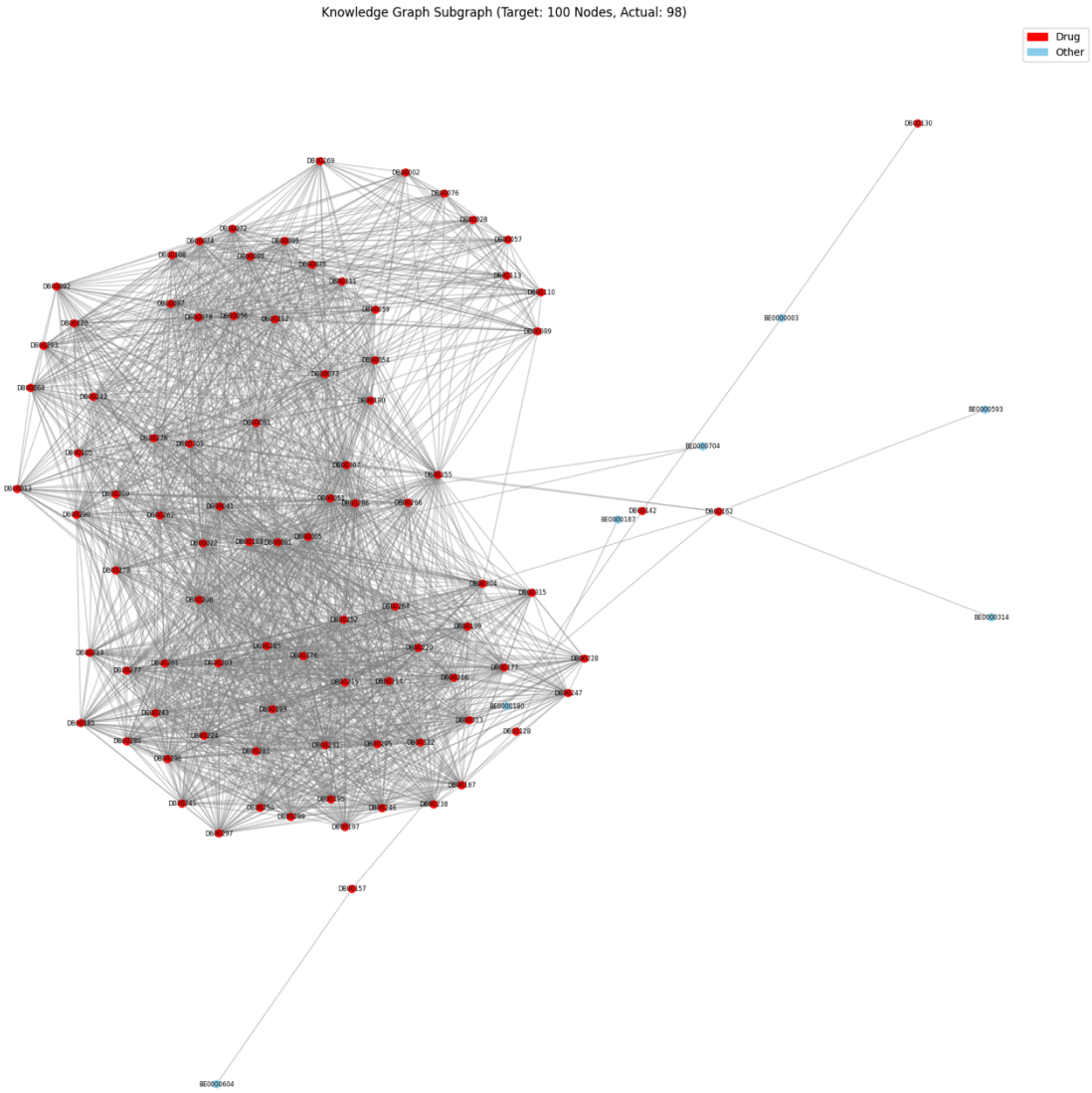
easy to explain the node types and the concept of relationships. It highlights that drugs are part of a larger network, not isolated entities.

Figure A2 – 100-Node Subgraph

A dense and highly connected region of the graph highlighting the complexity and clustering of drug interactions at larger scales.

Target: 100 Nodes (Actual: 98 Nodes)

a. Visual Description: This subgraph is the most visually dense among all samples. A large central cluster of red drug nodes dominates the graph, with many overlapping edges forming a tightly packed structure. Most of the edges are between red nodes, showing strong drug-drug connectivity. A few light blue 'other' nodes are still visible, mainly around the edges of the cluster. These blue nodes connect to the drug nodes but are fewer in number. The labels inside the dense center are mostly hidden due to the high edge density, and the overall structure appears very compact and crowded.



b. Interpretation: This visualization clearly shows the high level of interaction between drugs in the knowledge graph. The central cluster suggests that many drugs are related through direct DDIs or shared biological connections. The fewer 'other' nodes at the edges suggest that most interactions in this sample are direct drug-drug links or involve shared targets that are already densely connected. The layout highlights how the graph becomes harder to interpret visually as more nodes are added, but also shows the rich set of relationships the KGNN can learn from.

c. Significance: This subgraph highlights the complexity of the graph as more context is

added. The dense core of drug nodes emphasizes why a simple model cannot handle this level of connectivity effectively. It supports the need for a model like KGNN that can learn from indirect paths and complex neighborhood structures. This graph also shows that the model has to make sense of many overlapping connections, which justifies using graph-based learning to capture the deeper patterns in the data.

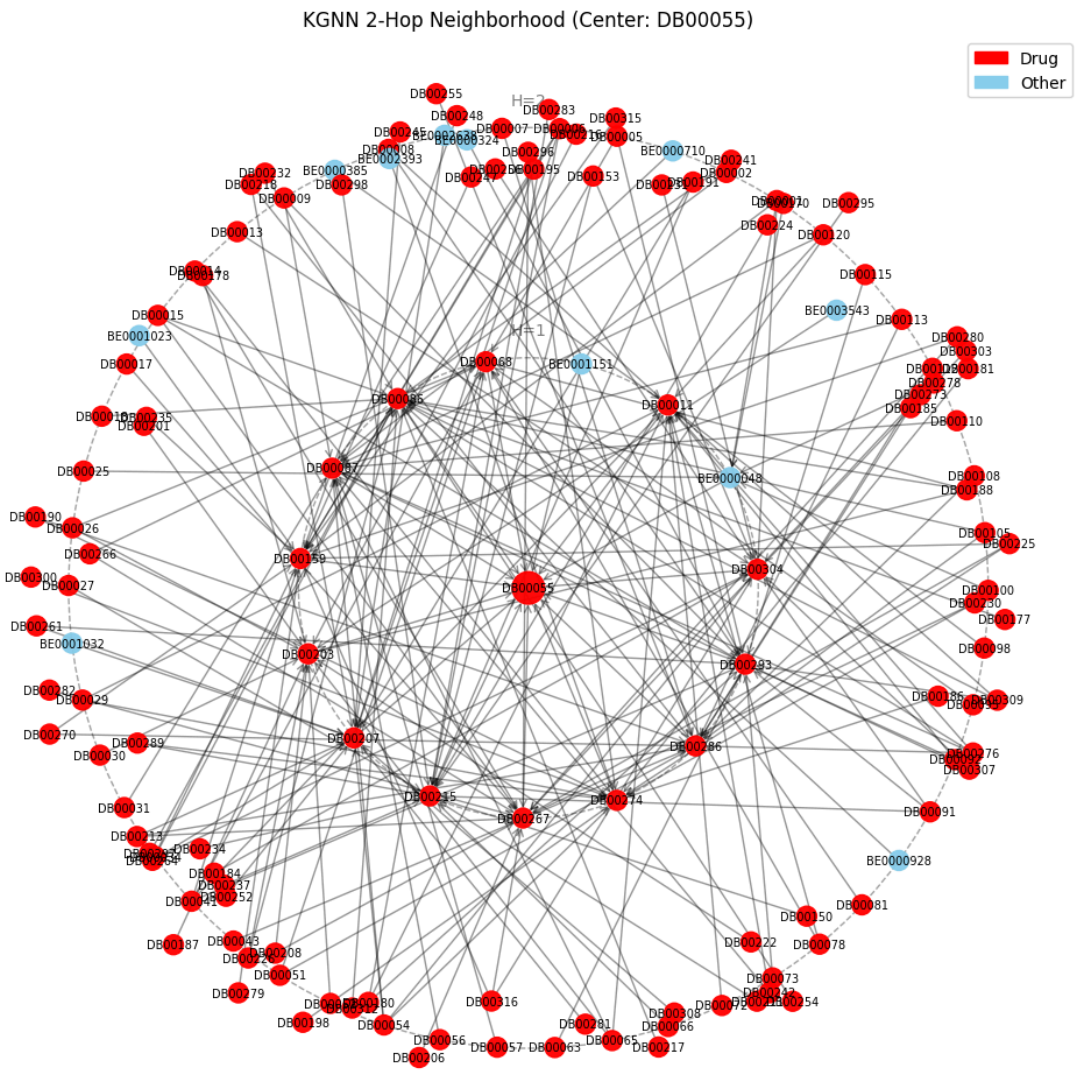
Appendix C: KGNN 2-Hop Neighborhood Visualization

Figure A3 – 2-Hop Neighborhood of Cyclosporine (DB00055)

A visualization of Cyclosporine's receptive field showing how the KGNN aggregates multi-hop information from both direct and indirect biological neighbors.

Center: Node 177 (Drug: DB00055 - Cyclosporine)

a. Neighborhood Size: H=1 has 15 neighbors; H=2 has 122 neighbors.
b. Visual Description: This neighborhood's size and density are comparable to Methotrexate (Node 167). Cyclosporine (center) connects to 15 direct neighbors, again mostly drugs (red). The H=2 layer is large (122 nodes) and predominantly composed of drugs, with a few 'other' entities visible. The high number of inward-pointing edges shows extensive information flow towards the center during aggregation.



c. Interpretation: Cyclosporine is another highly connected drug in this graph representation. Its large receptive field implies numerous direct and indirect relationships considered by the KGNN. The model aggregates signals from the 15 H=1 neighbors, which are themselves influenced by the 122 H=2 neighbors. This wide context is likely crucial for predicting Cyclosporine's interactions accurately, as it's known to have many DDIs often related to metabolic enzymes (which would be 'other' nodes).

d. Significance: This further illustrates the typical neighborhood structure encountered

by the KGNN for many drugs in the dataset – a mix of direct drug interactions and connections to other biological entities, amplified by the interactions of those neighbors. It highlights how the model leverages multi-hop information.