Faculty of Sciences Department of Computer Science

Preparatory work for the Master Thesis

Knowledge Graphs and Drug Repurposing

Author: Siddharth Sahay
Promoter: Prof. Tom Lenaerts
Advisors: Dr. Nassim Versbraegen, Inas Bosch

Academic year 2024–2025

Contents

Abbreviations

- **KG** – Knowledge Graph
- **KGE** – Knowledge Graph Embedding
- **GNN** – Graph Neural Network
- **GCN** – Graph Convolutional Network
- **R-GCN** – Relational Graph Convolutional Network
- **GAT** – Graph Attention Network
- **CNN** – Convolutional Neural Network
- **RNN** – Recurrent Neural Network
- **DNN** – Deep Neural Network
- **AE** – Autoencoder
- **SME** – Semantic Matching Energy
- **NTN** – Neural Tensor Network
- **SIGN** – Scalable Inceptive Graph Neural Network
- **SpMM** - Sparse-dense Matrix Multiplication
- **BPR** – Bayesian Personalised Ranking
- **LLM** – Large Language Model
- **MLM** – Masked Language Model
- **RL** – Reinforcement Learning (agent)
- **XAI** – eXplainable Artificial Intelligence
- **SVM** – Support Vector Machine
- **PPI** – Protein–Protein Interaction
- **DTI** – Drug–Target Interaction
- **MODA** – Mechanism Of Drug Action
- **ATC** – Anatomical Therapeutic Chemical
- **MeSH** – Medical Subject Headings
- **AUROC** – Area Under Receiver Operating Characteristic curve
- **AUPR** - Area Under Precision-Recall curve

- **MR** – Mean Rank
- **MRR** - Mean Reciprocal Rank
- **AMR** - Adjusted Mean Rank

Notations

- $\mathcal{G} = (\mathcal{E}, \mathcal{R})$ – A knowledge graph $\mathcal{G}$ consists of a set of entities $\mathcal{E}$ and relations $\mathcal{R}$.
- (h, r, t) – A triple is where h is the head entity, r is the relation, and t is the tail entity. Negative triples, which are not present in the KG are still valid.
- $\mathbb{R}^d$ - d-dimensional embedding space.
- $h, r, t \in \mathbb{R}^d$ – Entity and relation embeddings represented in d-dimensional real vector space.
- $h, r, t \in \mathbb{C}^d$ – In models like ComplEx, embeddings lie in the d-dimensional complex vector space
- $h^{\perp} = M_r h$ – Projection of entity h into relation-specific space in TransR (similarly $t^{\perp} = M_r t$).

1. Background

1.1 Objectives

The main idea of this paper is to explore the many ways we can utilise Knowledge Graphs (KGs) and Knowledge Graph Embeddings (KGEs) for the task of Drug Repurposing. Drug repurposing, which is also sometimes called drug repositioning, involves identifying new medical uses for existing drugs [1].

In the past decade, there has been a growing trend of using Knowledge Graphs (KG) to present biological and biomedical data in a structured form, which is a natural fit for data about drugs, diseases, proteins, and other biological entities since these are inherently interrelated and intertwined [2, 3].

Knowledge graphs have come up as powerful tools to integrate biomedical data into more understandable and structured forms [1, 4]. When these graphs are combined with embedding techniques, which are a type of representation learning, we get an even more expressive means of performing inference, link prediction and pattern discovery. The advantage of using embeddings is that easier to reason with for machine learning tools than actual KGs. Knowledge Graph Embeddings can help us investigate deeper, latent relationships which lie beyond the direct links between nodes. It's been shown that using KGEs can lead to highly accurate predictions that frequently outperform other approaches and using KGs directly [5].

This forms the perfect foundation for a task like drug repurposing, which, at its core, is about learning the hidden relationships between drug compounds and how diseases manifest in our bodies: genes, pathways, and broader physiological mechanisms. Ultimately, the goal is to identify existing drugs that can positively influence our complex biological systems and treat diseases beyond the original purpose of the drug.

Thinking about future work at this point, we will benchmark the existing methods of drug repurposing and identify the most promising methods. Afterwards, we will examine whether these methods are useful for rare diseases. Another objective is to evaluate the effectiveness of these methods in the oligogenic setting, where multiple mutant genes come together to cause a disease. This is discussed later in the "Future research directions" section.

Throughout the preparatory work done so far and for the upcoming thesis, I've received support and supervision by Prof. Tom Lenaerts as well as from Dr. Nassim Versbraegen and Inas Bosch.

2. Laying the groundwork

In this chapter, I will summarise some of the key KGE techniques and explain why these embedding methods are chosen to explore further in the next year. Afterwards, I briefly explain why drug repurposing using KGEs is worth pursuing.

The main idea of using embeddings is to provide us with features that can then be utilised for training and classification in the later stages of our machine learning pipeline.

2.1 Knowledge Graphs & Knowledge Graph Embeddings

A Knowledge Graph is a structured representation of knowledge, where entities - represented using nodes - are connected through relations - which are represented using edges [6]. More formally, a KG can be represented as a set of triples (h, r, t), where h (head) and t (tail) are entities, and r is the relation linking them. For an example, we could have h as Luke Skywalker, t as Darth Vader and the linking r as "son of". Relationships in a KG are often directional, e.g., the reverse triple (Darth Vader, son of, Luke Skywalker) would not mean the same thing and would represent a completely different (and incorrect) fact.

KGs are widely used in various applications, including search engines, recommender systems, and question answering [6].

Since KGs often provide an incomplete view of the actual knowledge they represent, we use Knowledge Graph Embeddings (KGEs) to "look deeper". KGEs are essentially low-dimensional vector representations of entities and relations that capture the local and global structure of the knowledge graph and represent its inherent structure [6]. Instead of operating directly on discrete symbols, KGEs encode entities and relations in a way that preserves structural and semantic properties, making it easier to predict missing links and perform reasoning tasks [1, 7]. In other words, embedding models are a form of representation learning, which help us better learn the knowledge represented in KGs, and thus also learn patterns that may not seem apparent from the outset in the direct structures via node links.

Embeddings translate nodes and entities into a vector space that is not understandable by humans. While it's technically possible to embed entities and relations in very low-dimensional spaces (e.g., 3 dimensions), these representations are generally insufficient for capturing the true complexity and richness of the data. In practice, embeddings typically lie in higher-dimensional spaces—commonly 100, 200, 500 or even more dimensions, which strikes a balance between expressiveness and computational efficiency. These higher dimensions allow the model to capture nuanced relational patterns and latent structures that may not be explicitly present in the graph itself. Much like in principal component analysis (PCA), "reducing" the graph to such an embedding space can surface hidden trends and facilitate tasks like link prediction or clustering [2, 1].

The effectiveness of a KGE model depends on how well it preserves the structure of the original knowledge graph while enabling efficient computation for downstream tasks. In the next section, we will describe briefly some common embedding methods.

2.2 Common embedding techniques

To create an embedding from an input knowledge graph, we generally follow different methods depending on the kind of embedding space we are aiming to achieve and also the problem statement. Below are the most common models.

2.2.1 Scoring function-based methods

Translational models

These methods all use a distance-based scoring function. The most noteworthy models are as below:-

- TransE [8, 6] represents entities and relations in the same vector space. It assumes that the relation acts like a "translation" from the head entity to the tail entity where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$. It models a translation such as:

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}. \tag{2.1}$$

  The advantage of TransE is that it is quite simple and computationally efficient, it has an average time complexity of $O(d)$, where d is the dimensionality of the entity embedding space. It performs well on one-to-one relations, but it performs badly for 1-to-N, N-to-1 and N-to-N relations [6, 9, 7, 10].

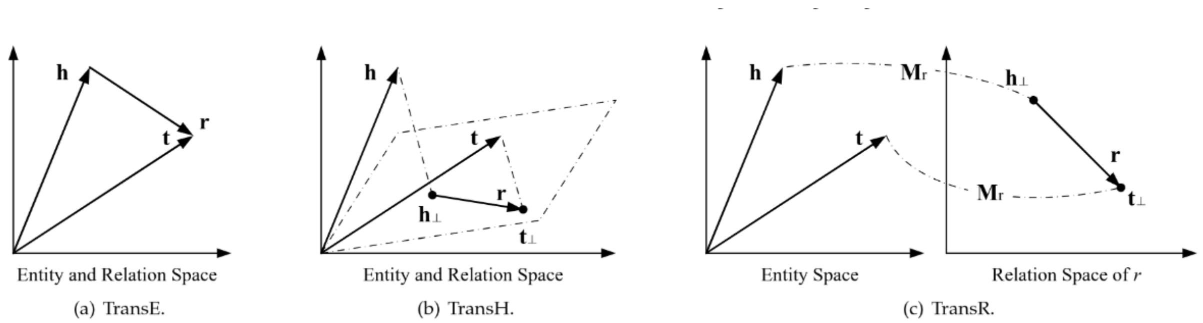


Figure 2.1: TransE, TransH and TransR. Figure from [1].

- TransR [6] extends TransE by introducing a relation-specific projection matrix to map entities into a relation-specific space.

$$\mathbf{h}_\perp = \mathbf{M}_r\mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r\mathbf{t}. \tag{2.2}$$

  The relation then operates in this transformed space:

$$\mathbf{h}_\perp + \mathbf{r} \approx \mathbf{t}_\perp. \tag{2.3}$$

- RotatE [6] represents relations as rotations in the complex vector space. Each entity is embedded as a complex vector, and each relation is modeled as an element-wise rotation from the head entity to the tail entity. Formally, for a triple (h, r, t), the model defines:

$$\mathbf{t} = \mathbf{h} \circ \mathbf{r}, \tag{2.4}$$

  where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$ are the complex embeddings of the head, relation, and tail respectively, and $\circ$ denotes the Hadamard (element-wise) product. Each component of $\mathbf{r}$ has a modulus of 1, i.e., $|\mathbf{r}_i| = 1$, ensuring that $\mathbf{r}$ represents a rotation in the complex plane. This formulation allows RotatE to model various relation patterns, including symmetry, antisymmetry, inversion, and composition [1, 6].

  RotatE is widely used for its efficiency and ability to capture diverse relational patterns.

In TransE, both entities and relations are in the same space, but with TransR we capture more information since each entity could be in multiple spaces where it implies different semantic meaning with respect to the relation and tail entity. It's much better at modelling multi-relational knowledge graphs, including 1-to-N, N-to-1, N-to-N relations. The one disadvantage of TransR is that it is computationally expensive, it has a average time complexity of $O(dk)$ operations for project per entity per relation [6, 7, 9, 10]. d represents the dimensionality of the entity embedding space and k is the dimensionality of the relation embedding space.

There are other variants as well that project entities onto relation-specific hyperplanes, use dynamic mapping matrices, or incorporate adaptive Mahalanobis distance for more flexible modelling [1]. The paper, by Ali et al [6], explores the effectiveness of straightforward techniques like TransE and TransR in the context of large biomedical KGs and achieving high performances, which implies a well-defined projection onto the embedding space [11]. Other studies also dive into using sparse matrix operations to accelerate the training of translational models [1, 6, 7]. By using the sparse-dense multiplication (SpMM), significant speedup in training times were achieved, as well as in memory usage [12]. These ideas make translational models a fair contender for application in large biomedical KGs. That said, from the translational models, TransE is still amongst the top-performing one just the way it is.

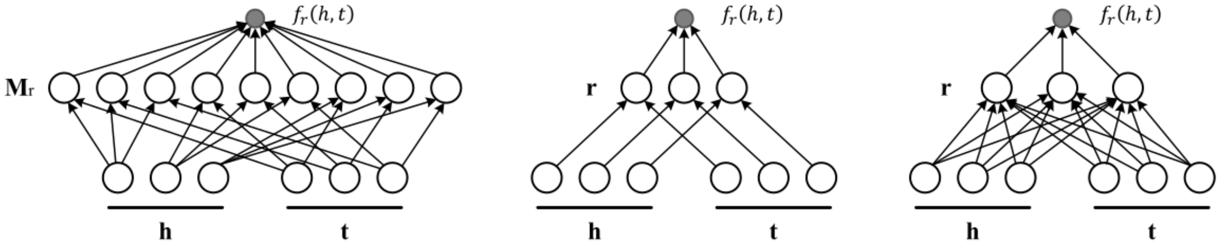
Bilinear models

- RESCAL was an early KGE model based on bilinear factorisation. Unlike the previously mentioned translational models, RESCAL assumes higher-order dependencies between entities. So

if A → B and B → C, then A → C, for example: (Brussels, LocatedIn, Belgium) + (Belgium, LocatedIn, Europe) then (Brussel, LocatedIn, Europe).
It also expresses symmetry better than translational models, for example (Luke Skywalker, SilblingOf, Princess Leia) then (Princess Leia, SiblingOf, Luke Skywalker) can be learnt easily.
Many KGE models fail to capture these higher-order patterns because they score each triple separately although they are very much linked [6, 7].

- DistMult is like a weighted dot product between entity embeddings. It assumes that every relation has a separate set of weights that interact with entity embeddings, but the interaction happens independently across each dimension [6, 9, 7]. By default, DistMult assumes all relations are symmetric. This is because it uses a diagonal matrix. Even though this seems like an issue, many of the relations in biomedical KGs are actually symmetric (for e.g., a relation like (compound causes side-effect) may seem like a non-symmetric relation, this is only because when expressed in language it appears to be so. From the perspective of holding knowledge about the interaction of the compound and side-effect, the relation is symmetric!)

- ComplEx extends the DistMult model by introducing complex-valued embeddings, allowing it to effectively model both symmetric and asymmetric relations. In ComplEx, entities and relations are represented as vectors in the complex space $\mathbb{C}^d$. Despite operating in the complex domain, ComplEx maintains computational efficiency and scalability, making it suitable for large-scale knowledge graphs [6, 9, 13].



(a) RESCAL. (b) DistMult. (c) HolE.

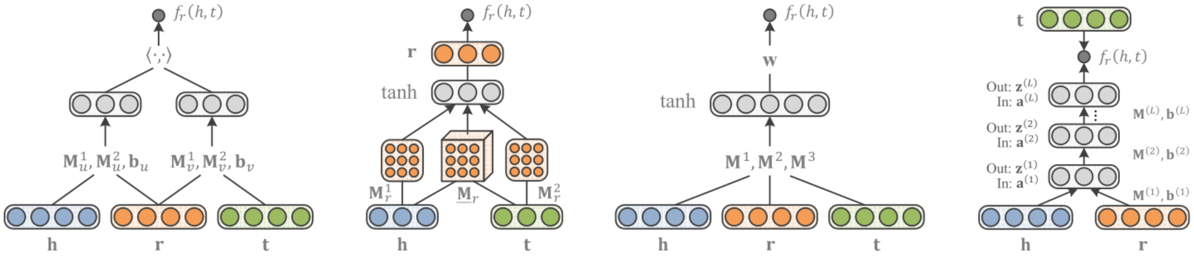
Figure 2.2: RESCAL, DistMulti, and HolE. Figure from [1].

An interesting development in this area is the exploration of multi-embedding interaction mechanisms, where entities and relations are represented by multiple embeddings to capture different aspects of their semantics. For example, a model might use separate embeddings to represent an entity's role as a subject and as an object. This approach can improve the model's ability to capture more complex relational patterns and has shown promise in some papers [2, 10, 14].

Neural network-based models

- ConvE is a knowledge graph embedding model that utilizes 2D convolutional neural networks to capture interactions between entities and relations [1, 6].

- R-GCN (Relational Graph Convolutional Network) extends traditional graph convolutional networks to handle multi-relational data, making it particularly suitable for knowledge graphs. R-GCNs are very effective in tasks such as link prediction and entity classification, especially in scenarios with highly multi-relational graph entities [9].



(a) SME. (b) NTN. (c) MLP. (d) NAM.

Figure 2.3: Neural network architectures of SME, NTN, MLP, and NAM. Figure from [1].

2.2.2 Path-based models

Random walk-based models

Random walk-based models generate sequences of nodes by simulating random traversals over the graph structure. Then these sequences are treated like sentences in natural language processing and are used to train models like Skip-gram to learn embeddings that capture both local and global structural information of the graph. If two entites appear together on many such walks, then it's implied that these entities are somehow related [7, 10, 15].

One of the biggest issues with random walk algorithms in the context of biomedical KGs is that here there is a densely connected protein-protein interaction (PPI) network [1, 7, 16]. We will see in a later section details about the algorithm DREAMwalk that addresses this by using semantic teleportation to escape the protein subgraph meaningfully [10].

- DeepWalk was one of the original random walk algorithms. It performs uniform random walks on graphs and applies the Skip-gram model to learn node embeddings, capturing community structures effectively [15].
- node2vec extends DeepWalk by introducing biased random walks that balance breadth-first and depth-first search strategies, allowing for more flexible exploration of node neighbourhoods [10, 15, 17].
- edge2vec extends node2vec by including edge semantics into the random walk process, enabling the model to capture both structural and relational information in heterogeneous graphs. node2vec and edge2vec are more performant on biomedical KGs due to the dense PPI network which makes regular DeepWalk ineffective [18].

2.2.3 Semantic matching model

- Semantic matching energy (SME) [19] maps entities and relations into a shared space and computes their compatibility through a learned transformation instead of direct dot products [6].
- Neural tensor network (NTN) extends SME by using a relation-specific tensor to capture complex interactions between entities, allowing richer representations [6, 20].

There are many other techniques for embedding, but for the sake of brevity, we will not go into the technical details here. For further details, one can review the paper by Ali et al [6]. The above are the most interesting approaches that are also the most widely used in the pipelines that follow in the next section about state-of-the-art.

2.3 Drug repurposing

Currently, there are about 7000 identified rare diseases, together affecting 10% of the population. However, fewer than 6% of these 7000 rare diseases have an approved treatment option, highlighting the great unmet needs in drug development [3, 21]. In most cases, we only have symptomatic treatment rather than cures. [21] The process of repurposing drugs for new indications, compared with the development of novel orphan drugs, is a time-saving and cost-efficient method [3]. It results in higher success rates, which can therefore drastically reduce the risk of drug development for rare diseases [16, 22].

Developing drugs for small patient populations is also not commercially viable for pharmaceutical companies. Orphan drugs are very expensive, e.g., Spinraza for Spinal Muscular Atrophy (SMA) costs $750,000 for the first year of treatment, followed by $375,000 for every subsequent year[21]. In general, traditional drug development is time-consuming (∼10-15 years) and costly (∼$2.5 billion per drug) [21]. In other sources, figures between $500 million and $2 billion are mentioned for drug development [13]. A part of this high cost comes from the fact that around 90% drugs fail during clinical development, with most failing due to side effects or adverse problems being discovered after passing phase I [22]. This creates a critical need for computational approaches that can accelerate discovery while reducing cost.

In this context, KGEs are as a powerful tool for systematic drug repurposing. Biological data is by nature heterogeneous and multi-relational. It links drugs, diseases, genes, proteins, pathways, and side effects. Representing this information as a knowledge graph allows us to model complex biomedical interactions at scale. But since manually/algorithmically identifying meaningful new connections directly from KGs is infeasible [1, 23]. KGEs address this by embedding entities and relations into a continuous vector space, enabling machine learning models to perform efficient link prediction, the main problem statement for drug repurposing. By learning from known drug–target or drug–disease associations, KGEs can learn about new drugs to repurpose for other diseases.

Benchmark studies demonstrate that KGE models such as DistMult and ComplEx outperform traditional matrix factorisation and graph-walk-based methods in drug–target interaction (DTI) prediction, polypharmacy side-effect analysis, and tissue-specific function inference [24, 20]. This approach, which combines the practical advantages of drug repurposing with the predictive power of KGEs, is a scalable, data-driven pathway to meet the unmet needs in rare disease treatment. This is a primary objective of my master thesis.

Repurposed drugs must still undergo phase II and III clinical trials for their new indication, which may reject 70% and 40% of compounds, respectively, for a variety of reasons [25] [26]. Even though developing drugs for new indications still requires a large investment and does not guarantee complete success, it can speed up or even help skip over phase I of drug develpoment. Considering the above reasons, I believe this is truly a noble pursuit.

3. State of the art

This chapter describes the specific techniques and algorithms that have emerged in the last 10 years for drug repurposing using knowledge graphs. Some of these are not directly posed as drug repurposing problems but are still within the context of link prediction, and similar techniques can be applied to our main problem statement.

Drug repurposing has been explored using many different computational methods such as signature matching, molecular docking, matrix factorisation, and network-based approaches [20, 21]. However, signature matching and molecular docking methods depend heavily on having comprehensive data and precise structural information about the target genes, which may not always be available. Matrix factorisation models identify new drug–disease interactions by measuring the similarity between drugs and disease-causing viruses based on their molecular sequences. However, these methods are limited to comparing pairs and cannot capture interactions on a broader, global scale [27]. Network proximity-based methods predict drugs for a disease by calculating how close the drug's target genes are to the disease's target genes in a network [22, 28, 13]. However, they struggle to include additional useful information in the network, such as similarities between different drugs or diseases [29].

3.1 Machine learning techniques

The general schema of most ML pipelines is described below in Figure 3.1.

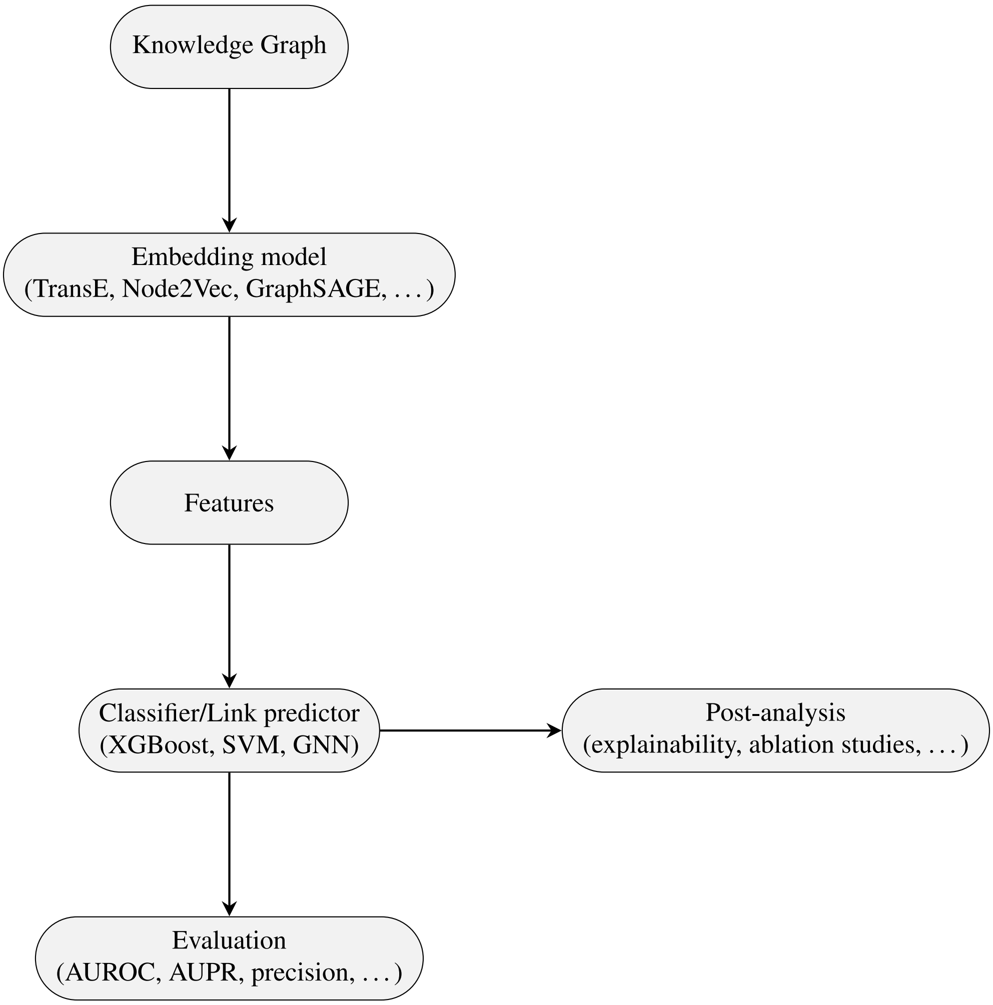

Knowledge Graph
Embedding model
(TransE, Node2Vec, GraphSAGE, . . .)
Features
Classifier/Link predictor
(XGBoost, SVM, GNN)
Post-analysis
(explainability, ablation studies, . . .)
Evaluation
(AUROC, AUPR, precision, . . .)

Figure 3.1: General ML pipeline for drug repurposing using knowledge graphs.

3.1.1 DT2Vec+

The DT2Vec+ [17] approach starts by integrating drug-drug and protein-protein similarity graphs with drug-disease and disease-protein graphs to create a three layer heterogeneous graph with all of these entity types.

For feature extraction, it uses node2vec as the KGE method into a 100-dimensional vector space. This was reported as the best threshold for accurately preserving graph information [17].

Classification was posed as a multi-label, multi-class classification task. The authors here used a gradient boosted tree approach, XGBoost. There were 6 total classifications the model was trained to predict: "increases-expression", "decreases-expression", "decreases-reaction", "increases-reaction", "increases-activity", "decreases-activity". These are the modes of action (MOA) of the drug. Categorical labels for the degree and type of interaction were converted to binary vectors through one-hot encoding, and one-vs-rest strategies were used to train the model against each label [17].

The drug-target interaction types were defined by concatenating pairs of drugs and target vectors extracted after the embedding stage. These were the inputs for the XGBoost classifier [17].

The testing and validation was performed using both internal and external test sets and a few different performance metrics were used. The authors used a 10-fold cross validation strategy and applied it to 90% internal data (training and validation combined). This means randomly dividing the data into 10 partitions and iteratively using 1 partition as the test set while training the model on the remaining 9. The remaining 10% of the data was kept completely separate as an external test set and used only for the final evaluation of the model's performance [17].

DT2Vec+ showed good results in predicting drug-target interaction for the 6 different interaction types (mentioned above). The accuracy was 77.09%, with an F1 score of 74.39% and precision of 84.58% [17].

3.1.2 ARBOCK

ARBOCK is not a technique for drug repurposing but it's approach is still relevant to the study here. It is a framework designed for predicting and interpreting disease-causing gene interactions. It uses the knowledge graph BOCK to achieve this [30].

BOCK was constructed with information collected from OLIDA [31] and other sources. It contains various node types, including gene, diseases, phenotype, biological process and edges such as co-expression, physical interaction, association with phenotype, etc. It does not however include drug related information, since it is more concerned with learning the interactions of genes rather than drug-target [30].

Initially, given a gene pair (which are already highlighted from being associated with oligogenic diseases), ARBOCK traverses all paths between the pair up to a length of 3. The paths are then abstracted into metapaths, which represent the sequence of nodes and edge types along the path. Then the association rule mining part of the algorithm begins: ARBOCK uses an adapted version of the Apriori algorithm to mine association rules from these metapaths. The rules help highlight the frequent patterns of metapaths that are characteristic of gene interactions in a pathogenic setting. The mined association rules are then used to train an interpretable decision set classifier [30].

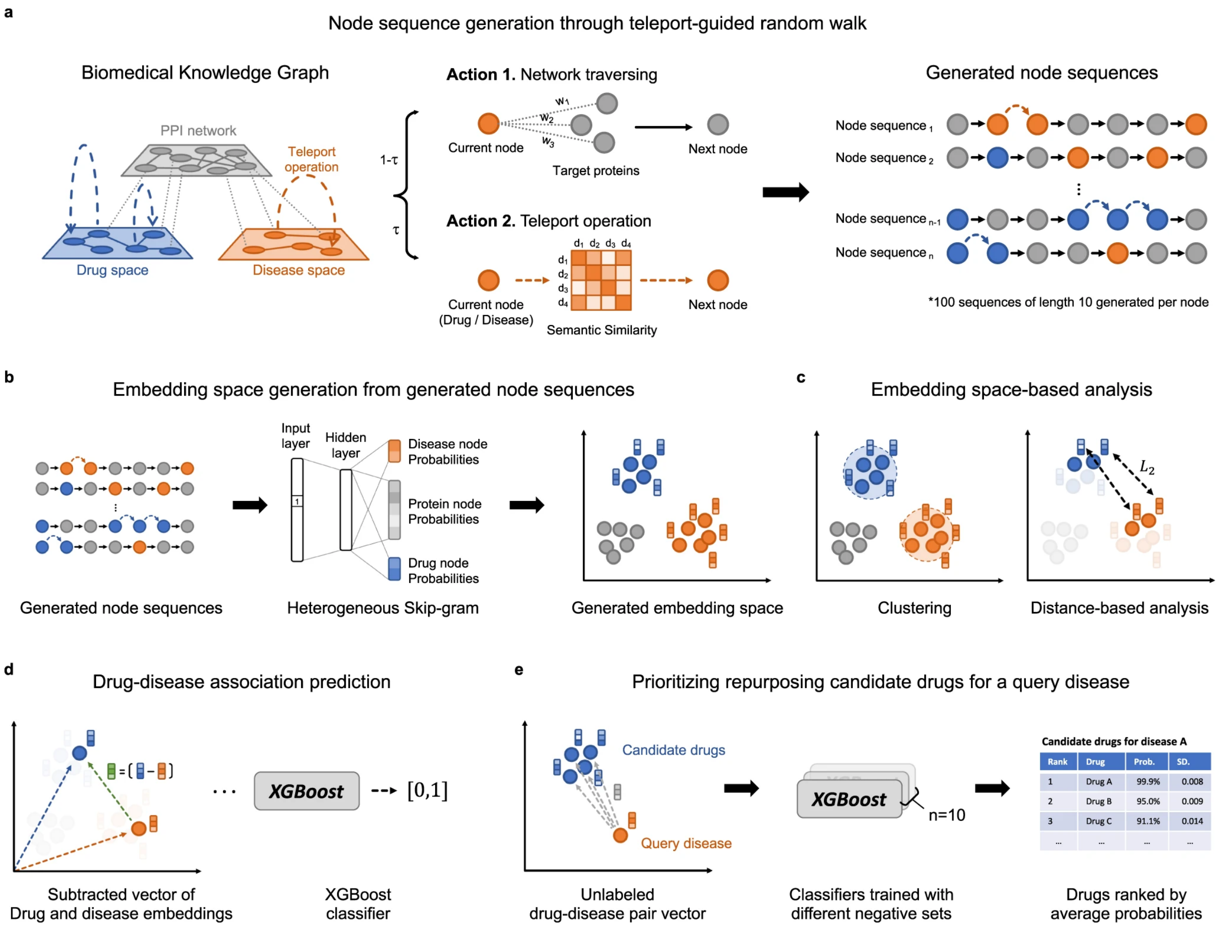
This approach is relevant to the drug repurposing problem since the understanding gained by ARBOCK about disease causing gene interactions can later be applied to the oligogenic setting of rare diseases. Once BOCK is supplemented with drug related information, it could even be used to find candidates for drug repurposing.

3.2 Random walk algorithms

3.2.1 DREAMwalk

This algorithm [10] uses an extension of a random walk approach we saw in the earlier chapter but improves upon it with semantic guided teleportation. The main problem with using biomedical KGs is that there is a dense PPI network, where it is easy for random walk algorithms to "get lost". DREAMwalk solves this by performing a random walk but when the walker lands on a drug or disease node, it has a probability to either continue traversing the network or teleport to another semantically similar drug or disease. Semantic similarity is calculated by using the ATC classification hierarchy for drugs, and MeSH, Disease Ontology or ICD-11 hierarchies for diseases [10]. This whole process generates the node sequences which are usually the output of a random walk algorithm.

Once this is done, the sequences are used to learn embedding vectors for the nodes using a heterogenous Skip-gram model. The goal here is to learn a continuous feature representation or embedding vector for each node in the network. This is achieved by trying to predict the surrounding nodes (the context) of a target node in the generated random walk sequences. Skip-gram models are often used to learn word


a
Node sequence generation through teleport-guided random walk
Biomedical Knowledge Graph
PPI network
Teleport operation
Drug space
Disease space
1-τ
τ
Action 1. Network traversing
w1
w2
w3
Current node
Target proteins
Next node
Action 2. Teleport operation
d1 d2 d3 d4
Current node (Drug / Disease)
Semantic Similarity
Next node
Generated node sequences
Node sequence 1
Node sequence 2
Node sequence n-1
Node sequence n
*100 sequences of length 10 generated per node
b
Embedding space generation from generated node sequences
Generated node sequences
Input layer
Hidden layer
Disease node Probabilities
Protein node Probabilities
Drug node Probabilities
Heterogeneous Skip-gram
Generated embedding space
c
Embedding space-based analysis
Clustering
L2
Distance-based analysis
d
Drug-disease association prediction
= (−)
XGBoost
[0,1]
Subtracted vector of Drug and disease embeddings
XGBoost classifier
e
Prioritizing repurposing candidate drugs for a query disease
Candidate drugs
Query disease
Unlabeled drug-disease pair vector
XGBoost
n=10
Classifiers trained with different negative sets
Candidate drugs for disease A
Drugs ranked by average probabilities

Rank	Drug	Prob.	SD.
1	Drug A	99.9%	0.008
2	Drug B	95.0%	0.009
3	Drug C	91.1%	0.014
...	...	...	...

Figure 3.2: DREAMwalk: a) The node sequence generation process through teleport-guided random walk. When arriving at a drug/disease node, the random walker selects an action between network traversing and teleport operation based on the teleport factor τ. b) The embedding space generation process with heterogeneous Skip-gram model. The heterogeneous Skip-gram performs negative sampling process from the same node types. c) The embedding vector space enables computational analysis including clustering of entities and distance-based analysis. d) Drug-disease association prediction using XGBoost classifier with subtracted vectors of drug and disease embedding vectors as input. e) Repurposing candidate drugs are prioritized using the trained XGBoost classifiers. Given a query disease of interest, all unlabeled drug-disease pair vectors are pass through the trained classifiers to obtain treatment probabilities. These probabilities are then averaged to yield a ranked list of candidate drugs based on their average treatment possibility. Figure taken from [10].

embeddings for natural language to predict next words, here the difference is that node types are different so the authors use a heterogenous Skip-gram instead. Since this model is type-aware, it performs negative sampling only from nodes of the same type.

Next, an XGBoost classifier is used for predicting the association of drug-disease links. The classifier takes the subtracted vector of drug and disease embeddings as input and outputs the drug-disease treatment probability.

A detailed overview of the entire framework is shown in Figure 3.2.

The benchmarks to compare the performance of DREAMwalk were prediction accuracy, area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPR). It was also compared against network-based models like NEWMIN and DTi2Vec [32] (different from DT2Vec+!), and GNN based models like SEAL and WalkPool. ComplEx and RotatE were also compared. The results of DREAMwalk outperformed all baseline models in random data splitting experiments across all three KGs that were used. On average over the three datasets, DREAMwalk achieved an accuracy of 87.3%, an AUROC of 93.8%, and an AUPR of 93.9%. This was better than NEWMIN (the best walk-based model) with an average accuracy of 84.0%, AUROC of 91.3%, and AUPR of 91.3%, and WalkPool (the best GNN-based model) with average accuracy of 76%, AUROC of 82.7%, and AUPR of 82.9%. ComplEx showed the best performance among transition-based approaches with an average accuracy of 70.6%, AUROC of 83.3%, and AUPR of 85.8% [10].

Ablation studies were also conducted to highlight the important of the semantic information-guided teleportation logic, and here as well DREAMwalk performed significantly better than a model without teleportation and with random teleportation.

DREAMwalk ultimately also proved itself useful by predicting associations for Alzheimer's disease drug candidates in phase 3 clinical trials. DREAMwalk achieved the highest median probability and rank for these candidates compared to the baseline models, showing its effectiveness in identifying promising drug repurposing candidates.

3.2.2 AnyBURL

AnyBURL [13] works in a different way than most of the other algorithms and techniques mentioned in this chapter. It's a bottom-up approach which first learns logical rules by performing random walks iteratively. Over a set time period, these random walk paths are sampled and logical rules are extracted from them. Each of these derives rules is then evaluated and assigned a confidence score based on it's accuracy to predict positive instances across all its inferences [13].

These learned rules are then used directly for link prediction/graph completion tasks. When the model is given a query with a head and a relation, it predicts the tail which has the highest confidence. The rule miner in AnyBURL can be used to initialise the generator in XG4Repo, which is a framework for explainable drug repurposing using KGs [13].

3.3 Deep learning techniques

The number of different architectures for deep learning models is vast [33, 34, 35, 36, 37]. To list a few, we have graph convolutional networks (GCN), network embeddings (NE) or knowledge graph embedding models (KGE), autoencoders (AE), fully connected deep neural networks (DNN), and lastly, recurrent neural networks (RNN) and convolutional neural networks (CNN) [27, 33]. These are all forms of representation learning; they can be broken down to the fact that deep learning is a special branch of machine learning where the multiple layers of non-linear processing units are used to extract high-level features from the raw data or input [15]. The performance of deep learning methods is directly linked to the effective data representation. The feature extraction and learning stage is intertwined with the model training process [38].

Even though many sources in the literature claim that deep learning models typically don't provide human-understandable explanations to support predictions [13, 39], there have been numerous post-

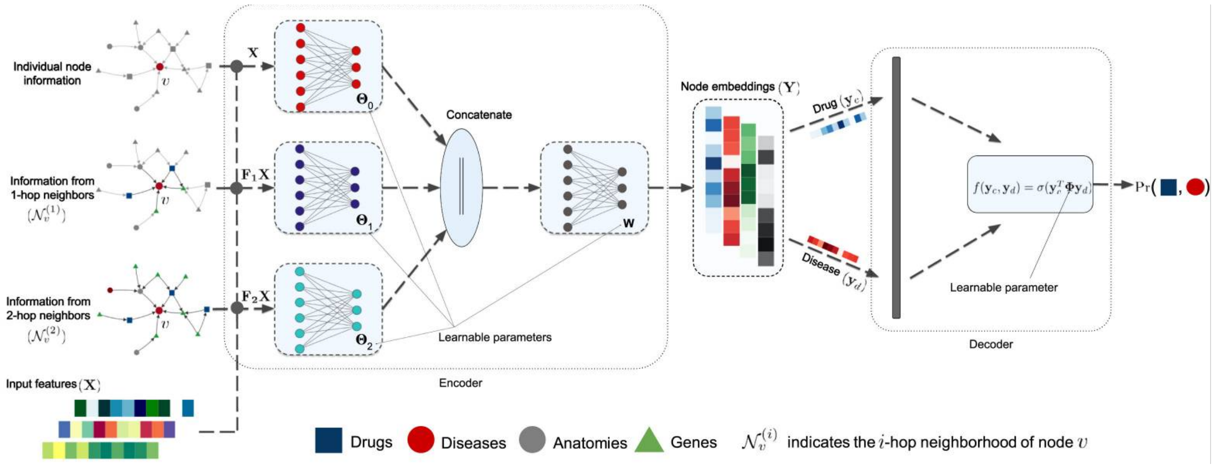

Individual node information
v
Information from 1-hop neighbors
$(\mathcal{N}_v^{(1)})$
Information from 2-hop neighbors
$(\mathcal{N}_v^{(2)})$
Input features $(\mathbf{X})$
$\mathbf{X}$
$\mathbf{F}_1\mathbf{X}$
$\mathbf{F}_2\mathbf{X}$
Θ_0
Θ_1
Θ_2
Concatenate
w
Learnable parameters
Encoder
Node embeddings $(\mathbf{Y})$
Drug $(\mathbf{y}_c)$
Disease $(\mathbf{y}_d)$
$f(\mathbf{y}_c, \mathbf{y}_d) = \sigma(\mathbf{y}_c^T \Phi \mathbf{y}_d)$
Learnable parameter
Decoder
Pr(■, ●)
■ Drugs
● Diseases
● Anatomies
▲ Genes
$\mathcal{N}_v^{(i)}$ indicates the i-hop neighborhood of node v

Figure 3.3: GDRnet architecture, figure from [29].

hoc methods to complement these techniques, which essentially complete the circuit and allow us to leverage the best of both worlds [4]. Additionally, there are other reinforcement learning methods such as MINERVA which can be trained to navigate the KG from a given source to a given destination yielding a path for the prediction of a model. It can further be augmented with a metapath strategy to rate certain routes through the KG higher than others [13]. KGML-xDTD [40] is another approach that uses an RL agent using the actor-critic architecture, which is useful in path-based explanation models [40]. These methods are, of course, classifier agnostic. We discuss explainability methods later on in a dedicated section.

A peculiarity with most of the deep learning models we discuss here is that they have an almost unreasonably good performance. 0.95+ AUROC and AUPRC. This is noteworthy and will be investigated as the first step of evaluating these techniques during benchmarking.

3.3.1 GDRnet

GDRnet [29, 37] is also a GNN-based model made for drug repurposing that frames the problem as a link prediction task. The model has an encoder-decoder architecture.

The approach begins by constructing a four-layered heterogeneous graph where nodes represent four biological entities (drugs, diseases, genes, and anatomies) and edges represent the various interactions between them. This network is made of almost 42,000 nodes and 1.4 million edges, so it contains both inter-layer connections (like drug-disease treatment relationships or drug-gene targeting) and intra-layer connections (like drug-drug similarities) [29]. This structure lets GDRnet make use of indirect relationships that can happen through shared gene targets or anatomical connections.

The first component, which is the encoder, is based on the Scalable Inceptive Graph Neural Network (SIGN) architecture, which precomputes neighbourhood aggregations before training to improve computational efficiency. This is particularly important when dealing with large biological networks, as it makes the runtime independent of the number of edges in the graph. The encoder generates low-dimensional embeddings for each node by aggregating information from its neighbourhood, incorporating data from up to 2-hop neighbours. These embeddings capture the structural connectivity information and entity-specific features [29].

The decoder then takes these embeddings and computes a score for each drug-disease pair using a quadratic norm scoring function. This function measures the correlation between drug and disease embeddings and produces a probability that a drug treats a particular disease. The model is trained using a weighted cross-entropy loss function to handle the class imbalance in drug-disease links [29].

For evaluation, GDRnet utilises several performance metrics. Classification performance is measured using AUROC and AUPRC. However, the more important metric is ranking performance, which assesses how well the model ranks known treatment drugs for diseases in the test set. The authors of the paper show that for a majority of diseases, GDRnet places the actual treatment drug within the top 15 recommendations, outperforming most other graph-based methods [29].

GDRnet is useful because it can quickly analyse large biological networks (unlike other approaches with GNNs). It works faster and more efficiently than older methods, making it suitable for real-world drug repurposing. The model showed good results during the COVID-19 pandemic, correctly predicting drugs like Dexamethasone, Ivermectin, and Sirolimus, which were later studied for their effects on the disease [29]. By capturing the complex interactions inherent in biological systems, GDRnet helps speed up the search for new treatments, saving both time and money [29].

3.3.2 DRAGNN

DRAGNN [34] is another GNN based model designed to predict new drug–disease links.

What makes DRAGNN different is how it uses attention to focus on the most useful information from a node's neighbours. Attention is represented through learnable coefficients that determine the importance of a node's neighbors during feature aggregation [41]. For e.g., if a drug is connected to many other drugs and diseases, DRAGNN learns to pay more attention to the ones that are more relevant, rather than treating them all the same. When updating a node's features, it ignores the node's own info and only uses its neighbours. This avoids the model just repeating what it already knows and forces it to learn from the graph [34].

After updating the features (embeddings) of all nodes, DRAGNN combines the drug and disease embeddings using a simple element-wise multiplication, and then passes that through a neural network to get a prediction score. This score is the probability of a drug treating a disease.

DRAGNN was tested on three public datasets and got strong results: AUROC around 0.947 and AUPR around 0.571 [34].

3.3.3 EKGDR

EKGDR [38] is a GNN-based system for drug repurposing. At a high level, it frames the problem as a recommendation task, where diseases correspond to "users", drugs correspond to "items", and the known associations between them are "interactions". EKGDR then supplements these known interactions with side-information in the form of a KG. This KG has a diverse range of nodes (e.g., drug entities, protein targets, side effects, molecular substructures, and more) and includes edges for known drug–drug interactions, drug–side effect relationships, drug classifications, etc. By including all these heterogeneous links, EKGDR aims to learn richer representations of drugs and diseases, and therefore improving repurposing accuracy.

EKGDR uses a multi-layer, relational path-aware GNN to embed nodes and relations in an end-to-end fashion. On the node side, each drug, disease, or other KG entity is assigned a learnable embedding vector. For relations, the system applies a rotation-based embedding mechanism (inspired by RotatE and ComplEx) that models each distinct edge type (like "binds to" or "causes") via a rotation in complex vector space. Unlike other node-based aggregation, EKGDR explicitly captures multi-hop paths (between 1 to 4 [38]) and the sequence of relations along those paths, so that higher-order neighbourhood information can influence the node representations. Additionally, it introduces the notion of "intents" behind disease–drug interactions: each intent is effectively a latent embedding that places attention-weighted emphasis on different relations in the KG. As the model is trained, it allocates varying degrees of importance to particular relation types to better reflect the observed disease–drug interaction patterns [38].

For classification, EKGDR uses a dot product between the final embedded vectors for the disease and the drug. This dot product score is then passed through a sigmoid-like function to produce an interaction probability. During training, the system uses Bayesian Personalised Ranking (BPR) loss, which helps it treat unobserved disease–drug pairs carefully—allowing the model to distinguish between genuinely negative associations and pairs that might still be unknown or missing from the data. The end-to-end GNN learning means that all embeddings (including diseases, drugs, and relations, as well as the attention parameters for each intent) get updated jointly as the model optimizes interaction prediction accuracy [38].

Performance-wise, EKGDR outperforms many baseline models on a dataset containing 6,657 known disease–drug interactions. Key evaluation metrics include the AUROC, AUPRC, and recall at rank K (recall@K). EKGDR achieves an AUROC around 0.9475, AUPRC of approximately 0.9490, and recall@200 of roughly 0.8315, which beats earlier methods such as deepDR and other knowledge graph–based approaches. Also, there were case studies on Alzheimer's and Parkinson's diseases which show that EKGDR's top-ranked drug suggestions align with experimental evidence [38]. This shows that it has the capacity to make meaningful and biologically relevant predictions.

There have also been further improvements suggested with pre-training and recommendation system approaches with another deep learning framework called UKEDR to further improve performance [36].

3.3.4 DTD-GNN

DTD-GNN [35] is (yet another) graph neural network designed for drug repurposing. The key innovation here is the introduction of event nodes that represent a complete drug–target–disease triplet, instead of modelling only pairwise interactions. This allows DTD-GNN to capture more biologically meaningful relationships and better reflect the multi-entity nature of real-world pharmacology.

The model constructs a heterogeneous graph with four types of nodes: drugs, targets, diseases, and the event nodes. Each event node connects to its corresponding drug, target, and disease. To embed these nodes effectively, DTD-GNN uses a combination of GCNs for aggregating neighbour information and Graph Attention Networks (GAT) to assign different weights to different neighbour nodes. This attention mechanism helps the model focus on the most relevant biological signals and filter out noisy or redundant connections. A gating unit further refines the information flow before making predictions [35].

DTD-GNN achieved outstanding results with AUROC of 0.987, AUPR of 0.984, and precision of 0.980, outperforming standard GNN baselines like GCN, GraphSAGE, and GAT [35]. Its ability to model three-way interactions and focus attention on critical nodes makes it a powerful approach for discovering new drug–disease links with high confidence. This makes DTD-GNN particularly interesting in my study of drug repurposing.

These results from DTD-GNN are extremely high and indicate a possible lapse in evaluation or overfitting.

3.3.5 Some notes about GNNs

Recent work has moved beyond vanilla GCNs as we can see in this section. There are many variants of GNNs that could not be discussed in detail. The survey paper, [37], goes into a detailed comparison and comments on the general deep learning landscape with respect to drug repurposing.

Graph attention networks, bilinear-attention blocks and heterogeneous graph transformers all reweigh neighbour messages so that the resulting embedding is more dominated by pharmacologically relevant nodes or relations. There are models such as DRGBCN, MGATRx and HGTDR [37] that consistently outperform regular GCN baseline models because they get rid of noisy neighbours (e.g. distant side-effect links) while amplifying the paths that actually explain a drug–disease interaction [37].

Another idea here is to use "disentangled" or variational representations. VGAEDR, EKGDR and related VGAEs [37] force the latent space to separate independent biological factors (chemical similarity, pathway overlap, target proximity). This not only improves link-prediction metrics, but also gives us more human-readable results. It can be helpful when we want to know whether a prediction is "structure-driven" or "network-driven" [15]. Yet another idea tackles the geometry of the space itself. Hyperbolic GNNs like HGNN-DR [37] embed nodes in negatively curved space, which is a natural fit for the tree-like hierarchies that dominate biomedical KGs (gene–protein–pathway–disease). Representations like this become more compact and distance metrics better represent ancestor–descendant relations, which in turn accelerates convergence and shrinks model size [37].

Another idea worth noting here is the rise of multimodal/data-fusion GNNs. There are systems such as BiFusion, DRHGCN, GCMM, MilGNet and STRGNN [37] that ingest several similarity graphs (chemical structure, therapeutic class, gene–expression, literature text) and learn either modality-specific encoders or a shared heterogeneous GNN before merging embeddings in a "fusion" layer. This design usually delivers higher Recall@K because a missed signal in one view (e.g., structural similarity) can be picked up by another (e.g., phenotype overlap) [37].

Together, these approaches form an interesting approach to the drug repurposing problem. These are all avenues of research that are still developing with respect to our problem statement and require more thorough analysis. This will be an area of research for my Master thesis as well. Early research already suggests that combining these ideas, i.e., attention inside a hyperbolic space or variational hypergraph encoders, is a promising path.

3.4 Large language models

3.4.1 DrugChat

DrugChat [7] is also as an interesting approach that could be adapted for these tasks. Although, it was not originally designed for direct link prediction or drug repurposing. DrugChat is a chatbot powered by an LLM that answers questions about drug compounds. It takes a molecule (represented as a SMILE graph) and a prompt as input. The molecule graph is processed by a pre-trained Graph Neural Network (GNN) to obtain an embedding, which is then adapted into a format understandable by the LLM [42]. The pre-trained LLM then uses the prompt and the modified embedding to generate an answer. DrugChat is trained end-to-end, with the GNN and LLM weights frozen, and only the adapter's parameters are trainable. The training data consists of question-answer pairs from curated databases like PubChem. This method offers the potential for human-understandable explanations, similar to other advanced chatbots [7].

3.4.2 MoCoSA

MoCoSA [7] combines a structural encoder (focusing on the graph structure using translational embeddings) and a description encoder (an LLM focusing on entity descriptions). This combined approach represents an interesting method for link prediction. It performed well on some datasets. It was not originally intended for use in drug repurposing but has been applied there. This would be an interesting avenue to explore, but there isn't much literature for it in relation to biomedical KGs [7].

3.4.3 LMKE

LMKE [7] uses a Masked Language Model (MLM) to perform link prediction on knowledge graphs. In this method, the LLM receives a head entity and a relationship (along with their descriptions) and attempts to predict the tail entity [43]. This approach has shown some good performance on the link prediction task. The primary issue here, as with all LLMs is the lack of evidence or reasoning especially in the drug repurposing case [7]. This model was also not intended for use in biomedical settings though.

3.5 Other techniques

The below techniques don't fall into neat categories like the aforementioned.

3.5.1 RPath

RPath [44] is a technique made for drug repurposing by reasoning over causal paths in a biological knowledge graph (KG). It integrates transcriptomic data from both drug-perturbed and disease-specific data. Unlike other methods, RPath uses causal reasoning and links in the KG. It mixes known biology with transciptomic experimental data to suggest how a drug might work. This helps RPath find useful drug-disease pairs and also explain why they might work [44].

First, it finds all short, loop-free paths between a drug and a disease in the KG. The KG has nodes for drugs, proteins, and diseases, and edges showing effects like activation or inhibition. These paths show all the ways a drug could affect the disease. Next, RPath adds drug-response data to these paths. It checks if the gene activity changes matches what the paths suggests should happen. Paths that match are kept and others are removed. Finally, it adds disease-response data to the remaining paths. It looks for paths where the disease changes go in the opposite direction of the drug changes. This means the drug may reverse the disease's effects. These paths are given higher importance. RPath focuses on drugs that not only follow expected biology, but also undo disease-related changes.

Essentially, it checks the paths that "reach up" from the disease and "reach down" from the drug and end up overlapping. The backbone of this approach is the use of transciptomic data.

Using causal reasoning with transcriptomic evidence, RPath is a great approach to drug discovery. The main problem with RPAth is the availability of transcriptomic data. The paper mentions that ultimately the working set of drug-disease pairs was less than 10 [44].

3.5.2 PoLo

PoLo [45] is a reinforcement learning based algorithm, it stand for Policy-guided walks with Logical rules.

PoLo is interesting because of how it learns to move through the graph. It trains an RL agent to take steps from one node to another, learning which paths lead to good results. But instead of walking around at random, PoLo gives the agent rules to follow — these are logical patterns based on known biological facts, also called metapaths, for e.g., "Compound causes Side-effect which is caused by Compound which treats Disease" [45]. These make the process of finding relevant paths easier. When the agent finds a path that follows one of these rules, it gets a bonus reward. This helps the agent focus on paths that actually make could actually occur in the biomedical sense.

PoLo was tested on Hetionet, one of the largest biomedical graphs. It performed better than many other popular methods at link prediction for drug repurposing. It outperformed models that use embeddings (like TransE or ComplEx), graph neural networks (like R-GCN), and even the initial method it was based on, MINERVA [45].

Another version of PoLo that only uses rule-based paths (called PoLo-pruned) did even better in some cases [45]. Another big plus is that PoLo shows the paths it used to make a prediction — this makes it easier for researchers to understand and trust the results.

PoLo unfortunately also has some issues: it heavily depends on having valid rules. In the study, 10 rules were taken but there are many others which have a higher length that were ignored. It also only works with cyclic rules, which means it can miss some relevant patterns. In biomedical graphs with highly connected nodes, the agent can get overwhelmed and take less useful paths [45]. For example, the agent could easily connect with any node through a highly connected anatomy node (nodes which are known to be the most connected in biomedical KGs). Also, for KGs other than Hetionet, other methods like AnyBURL perform better [45].

PoLo is a strong method for tasks like drug repurposing. It mixes logic and learning in an interesting way, and it gives clear explanations for its predictions. It does not however use KGEs for its purpose.

3.5.3 GNBR

Global Network of Biomedical Relationships (GNBR) [28] is a method to find new uses for existing drugs, especially for rare diseases, by using a large KG built from medical research papers. The graph itself is called GNBR, and it integrates data from over 28 million PubMed abstracts to connect drugs, diseases, and genes using themes like "treats," "causes," or "inhibits," and each connection has a confidence score based on how strong the evidence is in the literature. It uses NLP heavily to do this. It then uses these confidence scores directly in a machine learning model that turns the graph into a set of numeric embeddings. These embeddings capture patterns in the data, so the model can predict new drug-disease links that are likely to be true, even if they aren't mentioned directly in the literature [28].

What makes GNBR unique is that it is a combination of literature-scale NLP, graph learning, and uncertainty modelling to infer drug repurposing candidates without relying on curated databases. It doesn't just generate predictions, it explains them by tracing paths through the graph, identifying biological mediators (like shared genes or protein interactions), and validating findings through external PPI network analysis. The model showed good results (AUROC = 0.89) across 30 top-scoring novel drug-disease hypotheses, several of which were supported by literature or plausible mechanistic evidence. Its ability to generalise from indirect relationships and provide interpretable outputs marks a meaningful step forward in automating drug discovery for underserved rare disease populations [28].

3.6 Key biomedical KGs

Here, I briefly list publicly available KGs that have been used across the sources and would be candidates for future work in my Master thesis. The option to create KGs using multiple sources also exists. All the KGs are compared in a tabular manner in Table 3.1 below.

3.6.1 Hetionet

Hetionet [22] is one of the largest biomedical knowledge graph that is publicly available, it has also been used extensively for drug repurposing [45]. It was created by integrating data from 29 public biomedical databases, it has 47,031 nodes across 11 types and over 2.25 million relationships spanning 24 relation types. v1 was last updated in 2017 but it is still in use [7, 1]. It's publicly available via a Neo4j browser[1]. The schema is described in Figure 3.4.

It's ideal for use in drug repurposing although it is missing recent data.

3.6.2 PharMeBINet

PharMeBINet [7] is an evolution of Hetionet and integrates an additional public resources to the original 29 sources for Hetionet. It increases the number of nodes from 47,031 to a whopping 2,869,407. The number of relationships also go from 2,250,197 to 15,883,653. This was updated much more recently, in 2024. [7, 46]

3.6.3 GNBR

The Global Network of Biomedical Relationships (GNBR) [28] is a large, heterogeneous biomedical knowledge graph built from the abstracts of scientific literature [26, 28]. It connects drug, disease, and gene/protein entities using a limited set of semantic themes that describe their relationships. GNBR is made up of more than 130,000 entities and more than two million edges, each annotated with one or more semantic themes and an associated confidence score. Themes such as "Treatment" and "Inhibits cell growth (esp. cancer)" are particularly useful for drug repurposing efforts. Expanding GNBR to include full-text articles, beyond just abstracts, could further enhance its utility in uncovering novel biomedical insights. It was created in 2018 [7].

3.6.4 Bioteque

Bioteque [7] is a knowledge graph that streamlines biomedical data into pre-calculated embeddings, making it readily usable for downstream machine learning tasks [7].

It integrates data from over 150 public biomedical databases, encompassing 12 types of biological entities (such as genes, diseases, tissues, and cells) and 67 types of associations. The resource contains

[1] https://neo4j.het.io/browser

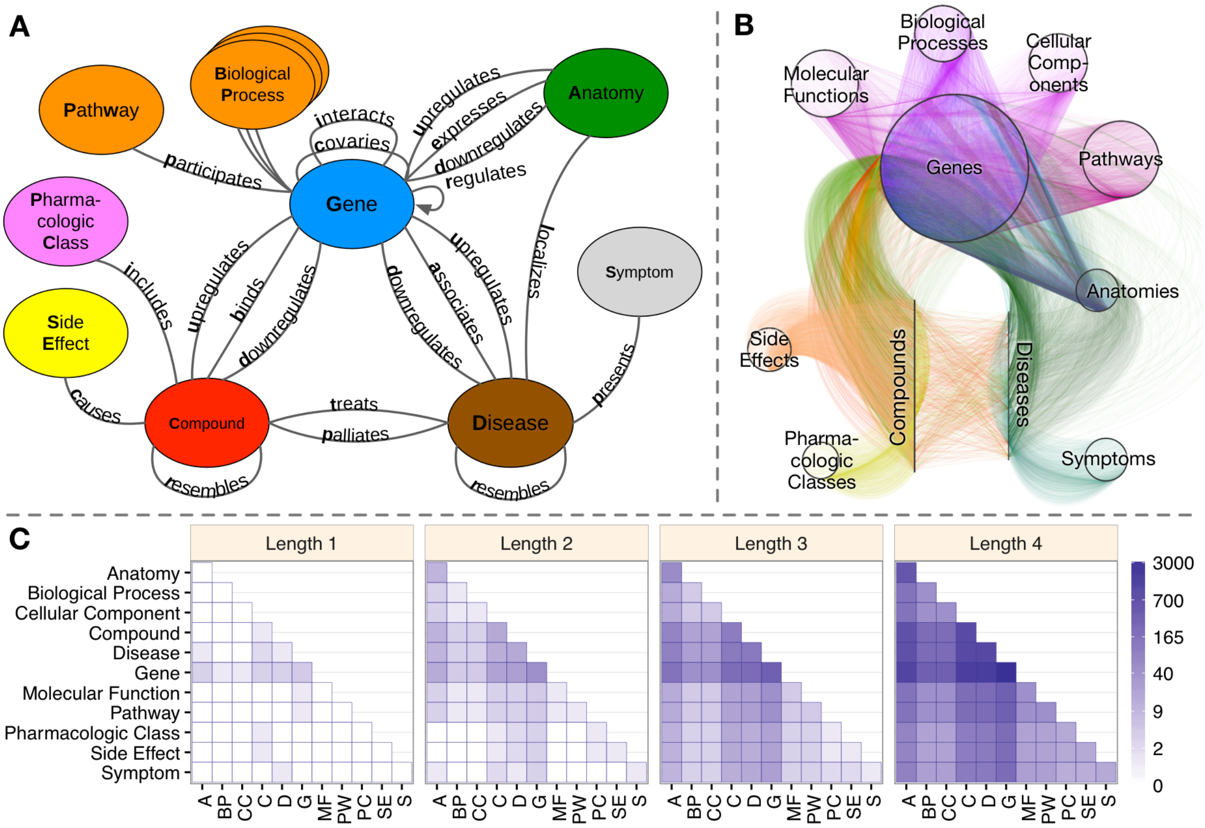


Figure 3.4: The Hetionet KG: A) The metagraph, a schema of the network types. B) The hetnet visualized. Nodes are drawn as dots and laid out orbitally, thus forming circles. Edges are colored by type. C) Metapath counts by path length. The number of different types of paths of a given length that connect two node types is shown. For example, the top-left tile in the Length 1 panel denotes that Anatomy nodes are not connected to themselves (i.e. no edges connect nodes of this type between themselves). However, the bottom-left tile of the Length 4 panel denotes that 88 types of length-four paths connect Symptom to Anatomy nodes. Figure taken from [22].

more than 450,000 biological entities and 30 million relationships between them. It was created in 2022 [7].

3.6.5 Clinical Knowledge Graph (CKG)

The Clinical Knowledge Graph (CKG) [7] stands out as one of the most comprehensive resources in the drug repurposing area [7]. It uses data from 26 biomedical databases and 9 ontologies to create a highly interconnected graph with 16 million nodes and over 220 million edges. The graph supports 19 node types—ranging from drugs and diseases to proteins and clinical variables—and 57 relationship types, enabling complex analyses. CKG is designed to support various analytical and machine learning applications for biomedical research.

CKG is actively maintained, it was recently updated in 2024. [7]

3.6.6 DRKG

DRKG [7] integrates data from six major biomedical databases, including DrugBank, Hetionet, STRING, IntAct, DGIdb, and GNBR. It also contains information from recent COVID-19-related publications.

It has 97,238 entities across 13 node types and contains about 5.87 million relationships spanning 107 relation types. It was created in 2020. [7]

3.6.7 BOCK

Although BOCK [30] does not yet contain drug information it comprehensively integrates disease-causing genetic interactions, which would be a perfect data source to the explore oligogenic setting. BOCK comprises 158,964 nodes of 10 different types. It contains 2,659,064 edges of 17 different types. [30]

3.6.8 OREGANO

This KG was introduced in 2023 and has 88,960 nodes across 11 distinct node types and 824,771 edges spanning 19 edge types. It was created to facilitate drug repurposing by integrating multiple data sources into one KG (much like many others listed here!). [7]

3.7 Interpretability & eXplainable AI

Interpretability of deep learning model predictions was briefly discussed in a previous section (although it was classifier agnostic); here, we will talk about general interpretability and XAI techniques. XAI stands for eXplainable AI, and refers to the idea of making AI systems transparent and understandable to humans. This transparency is especially crucial in biomedical settings, where it can build trust, ensure accountability and ethics.

3.7.1 Path-based reasoning

As previously mentioned, RL agents can be trained to navigate through KGs from a given drug to a target disease, giving us a path which explains the prediction in terms of it's interactions with genes or through pharmacology classes. MINERVA is one such approach [13]. KGML-xDTD [40] is another, which uses a two-step framework to first predict drug-disease scores with an embedding model and then uses graph reinforcement learning using the actor-critic architecture to find an interpretable path connecting the drug and the disease. KGML-xDTD leverages expert-curated "demonstration paths" (known mechanistic pathways from the literature) to guide the RL agent, making sure that the discovered paths are meaningful. This results in testable mechanisms of action (e.g. Drug X → binds Protein Y → regulates Pathway Z → influences Disease W) accompanying each prediction [40]. These are both post-hoc explanability methods.

Other methods, like XG4Repo [13] use metapaths instead of using RL agents to explore paths. Metapaths are sequences of entity types that form some meaningful connection patterns (like Drug-Gene-Disease, or Drug-SideEffect-Drug-Disease). XG4Repo creates these metapaths from the data and then makes predictions that are inherently explainable [13].

iDPath [47] incorporates path-based reasoning into the architecture of it's deep learning framework itself. It uses mechanisms of drug action (MODA) as paths and combines a graph convolutional network to capture global network context with an LSTM that simulates moving through a path from drug to disease [47]. iDPath also includes two attention models (which are known to increase interpretability) to highlight which intermediate entities (proteins or pathways) are the most influential on a given path. This way, iDPath can output an explicit ranked list of plausible paths explaining why a drug might treat a disease, and each path is annotated with the entities that influence the prediction [47].

3.7.2 Subgraph extraction

GNNExplainer [48] is a general method that finds a small subgraph (and feature subset) that was most influential for a particular GNN prediction. In our case of drug repurposing, we can apply GNNExplainer to a graph model that predicted a drug–disease link. It will try to select a subset of nodes and edges (for e.g., a small network of interactions linking the drug to disease) that most changes the prediction if removed [48].

Table 3.1: Comparison of key biomedical knowledge graphs

KG	Created or last updated	Nodes	Edges	Node types	Edge types	Data sources	Main uses
Hetionet v1.0 [22]	2017	47,031	2.25 million	11	24	29 public databases	Drug repurposing, disease-gene associations
PharMeBINet [7]	2024	2,869,407	15.88 million	66	208	Hetionet + 19 additional sources	Drug repurposing, disease-gene associations
GNBR [28]	2018	130,000+	2 million+	3	32 semantic themes	PubMed abstracts	Literature-based drug-disease-gene associations
Bioteque [7]	2022	450,000+	30 million+	12	67	150+ public databases	Precomputed embeddings
CKG [7]	2024 (2021)	20 million	220 million+	19	57	26 databases + 9 ontologies	Clinical and proteomics data integration
DRKG [7]	2020	97,238	5.87 million	13	107	6 databases + COVID-19 publications	Drug repurposing
BOCK [30]	2023	158,964	2.66 million	10	17	Curated cases + networks	Oligogenic disease gene interactions
OREGANO [7]	2023	88,960	824,771	11	19	7 biomedical databases	Drug repurposing with natural compounds

Another approach uses network perturbation to systematically remove certain connections to see if the prediction confidence drops, and uses this information to bring out nodes and paths that influence a prediction the most [49].

3.7.3 Logical pattern recognition

AnyBURL, as previously discussed, learns first-order logic rules from KGs in a bottom-up manner. This means that it samples numerous random paths in the KG and generalises them into the form “head → tail”, along with a confidence score. So a learned rule might be “drug X targets gene Y AND gene Y associates with disease Z → drug X treats disease Z”, which is another way of saying that if a drug X targets a gene Y involved in a disease Z, then the drug X may help treat the disease Z. This can serve as an explanation for “drug X treats disease Z” [13].

PoLo, which was discussed earlier as well, uses metapaths or logical rules. It uses these metapaths as a reward mechanism for the RL agent path explorer. If the agent correctly finds a path belonging to one of the metapaths defined manually then it’s rewarded [45]. This is similar to how MINERVA rewards it’s agent [13].

There have also been ensemble methods which create a set of rules that are mined after training embedding models to explain predicted links [50].

3.7.4 Attention interpretation

Graph attention networks (GAT) [41] and related models assign weights to edges or nodes in a neighbourhood. If a GNN predicts a link between drug X and disease Y, and uses attention, we can look at the attention weights on edges emanating from X (drug’s connections) and into Y (disease’s connections) during the prediction [41]. For example, an attention-based model might show that for drug X, the connection to protein A had a much higher weight than other connections, meaning that protein A was an important factor in linking X to disease Y [13].

3.7.5 Counterfactual reasoning

Similar to the network perturbation technique, counterfactual reasoning means adding or removing certain triples to see if a predicted drug-disease pair link still holds [49]. There are multiple ways to do this: additive, subtractive and contrastive [49]. Counterfactual explanations are quite powerful because they bring out actual causality, they indicate if the facts truly drive the prediction by testing the model’s dependency on them. In a field like drug repurposing, it’s invaluable to get a sense of adding/removing certain paths and knowing that the prediction fails. This approach often requires retraining or inference on perturbed inputs [49].

This is a computationally heavy approach since the search space grows exponentially as we add/remove incrementally, even while using heuristics. Yet, it is an effective way to verify results. The approach is closely related to Shapley values. [2]

3.8 Evaluation metrics

One of the key things to keep in mind while comparing different techniques are evaluation metrics. Below I briefly describe some common metrics that have been seen across the literature landscape for drug repurposing with KGs.

3.8.1 AUROC

AUROC is the Area Under the Receiver Operating Characteristic curve. It’s used to evaluate the performance of classification models in discriminating between positive and negative instances, and it can even

[2] https://en.wikipedia.org/wiki/Shapley_value

be used for multi-class classification. A higher AUROC score generally indicates better performance.

AUROC is quite valuable in drug repurposing tasks due to the nature of the data itself; there are much more non-associated drug-disease pairs than associated ones. A high AUROC indicates that the model effectively ranks true associations higher than false ones, which is crucial for prioritising candidates for actual experimental validation [1]. It can therefore be complemented with other metrics like the Area Under the Precision-Recall Curve (AUPR), which can help ensure the results are robust [10].

3.8.2 AUPR

Area Under the Precision-Recall Curve (AUPR) is also a crucial metric, especially because we are dealing with imbalanced datasets where true drug-disease associations are rare. The vast majority of possible interactions are unimportant [1, 7]. So, while AUROC can make a model look better than it really is, AUPR gives a clearer picture by focusing only on the correct predictions of real drug-disease connections.

Also, in drug repurposing, we care more about finding a few very promising drug candidates rather than getting a long list with lots of false leads. AUPR helps highlight the most accurate, top-ranked predictions, which is exactly what researchers need when deciding what to test in the lab [10, 21].

3.8.3 Hits@K

Hits@K is a metric that tells us about how often the correct answer shows up in the top K predictions made by a model. For example, Hits@10 means we're checking if the true drug-disease match is in the model's top 10 suggestions. The higher this number, the better the model is at putting the right answers near the top of the list. In drug repurposing, researchers usually care about finding a few strong candidates to test for themselves. Hits@K shows how often the model includes the correct drug in its top few guesses, which makes it easier to pick out the best ones for following up on [1].

Most drug-disease pairs in a KG aren't truly effective pairs, so the data is very imbalanced. Unlike accuracy, which can be misleading in such cases, Hits@K is better at showing how well the model finds the real matches.

Hits@K gives a generally clear picture of ranking performance, but it's often used with other metrics like Mean Reciprocal Rank (MRR) or AUPR, which give additional details [23].

3.8.4 Mean Rank (MR)

The Mean Rank computes the average rank of the correct entity across all test triples. Given a set of test triples $\mathcal{T}_{test}$, the MR is defined as:

$$\text{MR} = \frac{1}{|\mathcal{T}_{test}|} \sum_{(h,r,t)\in\mathcal{T}_{test}} \text{rank}(h, r, t), \tag{3.1}$$

where $\text{rank}(h, r, t)$ is the rank assigned to the correct entity in a ranked list of possible entities. A lower MR indicates better performance.

3.8.5 MRR

Mean Reciprocal Rank is also widely used for link prediction tasks with KGs. MRR assesses how well a model ranks the correct entity for a query among a set of candidate entities.

It's calculated with this formula:

$$\text{MRR} = \frac{1}{|K_{\text{test}}|} \sum_{(h,r,t)\in K_{\text{test}}} \frac{1}{\text{rank}(t)} \tag{3.2}$$

where K_{test} is the set of test triples, and $rank(t)$ is the rank of the correct tail entity t when the head entity h and relation r are given.

A higher MRR score means better performance. A model with an MRR of 1 would mean the model predicted every missing entity as the top result.

3.8.6 Adjusted Mean Rank (AMR)

The Adjusted Mean Rank normalizes the MR by the expected rank under random scoring:

$$\mathrm{AMR} = \frac{\mathrm{MR}}{\frac{1}{2}\sum_{(h,r,t)\in\mathcal{T}_{test}}(\xi(h,r,t)+1)}, \tag{3.3}$$

where $\xi(h,r,t)$ denotes the number of candidate triples against which the true triple is ranked. The AMR helps compare different datasets by adjusting for dataset size.

4. Limitations

There are several issues that come up while tackling the drug repurposing problem using knowledge graphs, which justify more sophisticated KGE architectures and data pre-processing steps. There are also issues with KGEs since we inherently lose some information when data is projected onto the embedding space, despite high dimensionality. Some of the these issues are described in this section.

4.1 Bias towards PPI

This is one of the biggest problem when dealing with large biomedical KGs. A common structure of these graphs is the dense protein-protein (or gene-gene) interaction networks, which are connected to sparse networks like drug-gene and disease-gene [26]. It can even be seen clearly on the Figure 3.4 which displays the Hetionet KG from a bird's eye view. Since the protein-protein network is so much more dense than the other parts of the KG, sometimes covering over 90% of nodes and edges [10], approaches like random-walk based embedding methods and even others tend to learn features from the gene-gene interactions, which is not representative of the entire graph. This bias can hinder the ability to find the associations we are trying to find.

Approaches like DREAMwalk employ a teleportation to avoid this [10]. The semantic information guided teleportation helps in escaping the PPI network. When the random walker lands on a drug or disease node, it can with a predefined probability teleport to another drug or disease node that is semantically similar, which helps in easing the bias towards gene-gene interactions.

4.2 Data incompleteness

KGs are constructed using data from various sources, but as we've seen from the sheer number of attempts there have been at creating a complete KG, it's often difficult to incorporate everything. Schemas are often limiting and inherently simplify some biological interactions in one way or another [7]. This leads to an incompleteness of data for drug repurposing. Data is also constantly outdated as new research is conducted, even KGs that are derived from text mining can be sparse and can contain unreliable or noisy data [28]. Another issue is accessing only publicly available literature, which can also contribute to incompleteness, as a significant portion of biomedical relationships might reside only in controlled-access publications [26].

This is why using the right KG and even considering combining multiple sources for a truer representation of the real-world is important. For rare diseases, this is even more true, since data is often scarcer and more distributed, making KG construction challenging and amplifying data incompleteness [39].

4.3 Scalability

KGE models are usually seen as more scalable than older graph analysis methods like path exploration [2]. However, as biomedical knowledge graphs grow larger and more complex, they still face big challenges. For e.g., graphs like the Clinical Knowledge Graph has millions of nodes and hundreds of millions of connections, which takes a lot of computing power [7]. Training advanced KGE models, especially GNNs, on these massive graphs takes a lot of resources. It often needs distributed computing, parallel processing, and special hardware like GPUs. Also, training deep GNNs that capture information from many hops away is tough because of computing limits and problems like oversmoothing, where the model loses useful details [15].

4.4 Beyond second-order neighborhoods

One of the main problems with traditional embedding methods is that they preserve immediate neighborhood structure, but for drug repurposing we are trying to rely on the indirect connections and reasoning across multiple hops. E.g., a drug might affect a disease by targeting a protein in a pathway that is associated with the disease, a path of length two or more [45].

4.5 Interpretability

Another big challenge in using KGEs is that they often lack transparency. Like many deep learning models, KGE methods can be "black boxes" which are not human-understandable, and make predictions without clearly showing how they got there. Even if these models are accurate, medical experts need understandable explanations to trust and check the results [13, 39]. This is why many methods have interpretability in the foreground, since it's of great important especially in biomedical related fields.

There are many post-hoc methods that exist to give relief here [13, 50, 48, 39]. We also have RL agents that can be trained to traverse graphs given a drug-disease pair to find a justification [45]. Some types of GNNs are designed from the start to be more explainable [15]. For example, attention mechanisms can learn which nearby nodes or paths are most important for a prediction. This helps show which parts of the graph had the biggest influence. These built-in methods for making models easier to understand are called ante hoc explainability techniques [37].

5. Future research directions

In this section, we go into details about what directions my Master thesis could take next year.

5.1 Benchmarking

There is a lack of benchmarking of the various approaches and techniques for drug repurposing using KGs and KGEs across the literature. The most recent review paper, [7], seems to compare only KGEs with gaps in the metrics and does so in an incomplete manner. Many of the KGs used here are not relevant to the drug repurposing problem statement. Moreover, the reported performance is collected from the source papers rather than reproducing and analysing original results consistently across the same KGs and metrics. In the paper by Sameh K et al [2], there was some benchmarking on KGE models as well, but it has been done for link prediction rather than specifically for drug repurposing as well.

In other words, there is a gap in the literature with regards to a more comprehensive benchmarking of all the machine learning, random walk, and deep learning based methods. Without such a comprehensive comparison, it would be hard to conduct further research in one area or another seeing as it hasn't been established which approach is objectively the best.

Many of the deep learning techniques report extremely high performance on the AUROC and AUPR metrics, 0.95+ in both [38, 36, 35]. This potentially suggests either overfitting or mis-evaluation. It would be prudent to verify the results before either dismissing these claims or further pursuing these approaches to fine-tune the pipelines and investigate avenues for improvement.

Another reason this gap exists is that drug repurposing is a fairly nascent field, with most of the boom in research happening in the last 5 years. So techniques like GNNs are still being explored extensively, and innovation is still ongoing [37].

Methods for benchmarking

In each stage of the pipeline shown earlier in Figure 3.1, we can evaluate different choices for models and techniques, including the KG itself, the KGE method, and the classification model. Initially, it would be beneficial to establish a comprehensive baseline comparison of the simpler and more straightforward embedding techniques, before we introduce complexities to the pipeline. Starting with some scoring function-based models like TransE, DistMult/ComplEx would provide a good understanding of their efficacy in the task of drug repurposing. Other embedding methods like ConvE can also be considered. Dimension sizes of 50, 100 and 200 can be used here since these are the norms for training generally [51].

Next, we could look at promising methods from the initial benchmarking, for e.g., random walk-based methods, like DREAMwalk. The initial investigations here can be simpler techniques and we can incrementally add complexity to improve performance. This will also highlight which additions yield better results, like a reverse ablation method.

For the classification tasks, we can explore and benchmark SVMs, MLPs, GNNs, XGBoost/random forest-based classifiers. The comparative analysis here will provide fodder for further investigations.

We can also investigate hyperparameter tuning and other optimisations with these baseline models. Once this benchmarking is complete, we could use these results and propose further improvements to existing methods or propose new methods altogether for our problem of drug repurposing. These could include incorporating metapaths, attention mechanisms, or improving explainability.

5.2 Explore the oligogenic setting

Another avenue to explore is applying these techniques to the oligogenic setting. Most of the papers discussed in previous sections focus on monogenic disease, but we know that rare diseases often manifest in an oligogenic setting [52]. To this end, BOCK could be augmented with drug data and used as the primary KG.

5.3 Conclusions

Ultimately, the initial benchmarking results will determine many of the future research directions for the Master's thesis. There is also plenty of space to introduce innovations with existing methods by improving robustness, performance and explainability.

Bibliography

[1] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. 2017.

[2] Sameh K Mohamed, Aayah Nounu, and Vít Nováček. Biological applications of knowledge graph embedding models. 2020.

[3] AH Jonker, D O'Connor, M Cavaller-Bellaubi, C Fetro, M Gogou, PACT Hoen, M de Kort, H Stone, N Valentine, and AMG Pasmooij. Drug repurposing for rare: progress and opportunities for the rare disease community. 2024.

[4] Elif Ozkan, Remzi Celebi, Arif Yilmaz, Vincent Emonet, and Michel Dumontier. Generating knowledge graph based explanations for drug repurposing predictions. 2023.

[5] Edoardo Ramalli, Alberto Parravicini, Guido Walter Di Donato, Mirko Salaris, Céline Hudelot, and Marco Domenico Santambrogio. Demystifying drug repurposing domain comprehension with knowledge graph embedding. 2021.

[6] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. 2021.

[7] Pablo Perdomo-Quinteiro and Alberto Belmonte-Hernández. Knowledge graphs for drug repurposing: a review of databases and methods. 2024.

[8] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. 2013.

[9] Kairong Hu, Hai Liu, Choujun Zhan, Yong Tang, and Tianyong Hao. Learning knowledge graph embedding with a bi-directional relation encoding network and a convolutional autoencoder decoding network. 2021.

[10] Dongmin Bang, Sangsoo Lim, Sangseon Lee, and Sun Kim. Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers. 2023.

[11] Chang Su, Yu Hou, Michael Levin, Rui Zhang, and Fei Wang. Protocol to implement a computational pipeline for biomedical discovery based on a biomedical knowledge graph. 2023.

[12] Md Saidul Hoque Anik and Ariful Azad. Sparsetransx: Efficient training of translation-based knowledge graph embeddings using sparse matrix operations. 2025.

[13] Ana Jiménez, María José Merino, Juan Parras, and Santiago Zazo. Explainable drug repurposing via path based knowledge graph completion. 2024.

[14] Hung Nghiep Tran and Atsuhiro Takasu. Analyzing knowledge graph embedding methods from a multi-embedding interaction perspective. 2023.

[15] Ruth Johnson, Michelle M. Li, Ayush Noori, Owen Queen, and Marinka Zitnik. Graph artificial intelligence in medicine. 2024.

[16] Nan Li, Zhihao Yang, Jian Wang, and Hongfei Lin. Drug–target interaction prediction using knowledge graph embedding. 2024.

[17] E. Amiri Souri, A. Chenoweth, S. N. Karagiannis, and S. Tsoka. Drug repurposing and prediction of multiple interaction types via graph embedding. 2023.

[18] Zheng Gao, Gang Fu, Chunping Ouyang, Satoshi Tsutsui, Xiaozhong Liu, Jeremy Yang, Christopher Gessner, Brian Foote, David Wild, Ying Ding, and Qi Yu. edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. 2019.

[19] Xavier Glorot, Antoine Bordes, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data, 2013.

[20] Xiangxiang Zeng, Xinqi Tu, Yuansheng Liu, Xiangzheng Fu, and Yansen Su. Toward better drug discovery with knowledge graph - sciencedirect. 2022.

[21] Helen I Roessler, Nine V A M Knoers, Mieke M van Haelst, and Gijs van Haaften. Drug repurposing for rare diseases. 2021.

[22] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Greena, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. 2017.

[23] Zhenxiang Gao, Pingjian Ding, and Rong Xu. Kg-predict: A knowledge graph computational framework for drug repurposing. 2022.

[24] Rivas-Barragan Daniel, Domingo-Fernández Daniel, Gadiya Yojana, and Healey David. Ensembles of knowledge graph embedding models improve predictions for drug discovery. 2022.

[25] Shabunina EA, Malyshev I Yu, Kuznetsova LV, and Lobanov EV. Evolution of the drug repurposing paradigm. 2021.

[26] Anton Yuryev, Maria Shkrob, Alex Tropsha, and Grant Mitchell. Exploring drug repurposing for rare diseases: Leveraging biomedical knowledge graphs and access to scientific literature. 2025.

[27] Xiaoqin Pan, Xuan Lin, Dongsheng Cao, Xiangxiang Zeng, Philip S. Yu, Lifang He, Ruth Nussinov, and Feixiong Cheng. Deep learning for drug repurposing: Methods, databases, and applications. 2022.

[28] Daniel N Sosa, Alexander Derry, Margaret Guo, Eric Wei, Connor Brinton, and Russ B Altman. A literature-based knowledge graph embedding method for identifying drug repurposing opportunities in rare diseases. 2020.

[29] Siddhant Doshi and Sundeep Prabhakar Chepuri. A computational approach to drug repurposing using graph neural networks. 2024.

[30] Alexandre Renaux, Chloé Terwagne, Michael Cochez, Ilaria Tiddi, Ann Nowé, and Tom Lenaerts. A knowledge graph approach to predict and interpret disease-causing gene interactions. 2023.

[31] Charlotte Nachtegael, Barbara Gravel, Arnau Dillen, Guillaume Smits, Ann Nowé, Sofia Papadimitriou, and Tom Lenaerts. Scaling up oligogenic diseases research with olida: the oligogenic diseases database, 2022.

[32] Maha A. Thafar, Rawan S. Olayan, Somayah Albaradei, Vladimir B. Bajic, Takashi Gojobori, Magbubah Essack, and Xin Gao. Dti2vec: Drug–target interaction prediction using network embedding and ensemble learning. 2021.

[33] Xiaorui Su, Lun Hu, Zhuhong You, Pengwei Hu, Lei Wang, and Bowei Zhao. Deep learning method for repurposing antiviral drugs against new viruses via multi-view nonnegative matrix factorization and its application to sars-cov-2. 2021.

[34] Yajie Meng, Yi Wang, Junlin Xu, Changcheng Lu, Xianfang Tang, Tao Peng, Bengong Zhang, Geng Tian, and Jialiang Yang. Drug repositioning based on weighted local information augmented graph neural network. 2023.

[35] Wenjun Li, Wanjun Ma, Mengyun Yang, and Xiwei Tang. Drug repurposing based on the dtd-gnn graph neural network: revealing the relationships among drugs, targets and diseases. 2024.

[36] Dongsheng Cao, Kun Li, Jiacai Yi, Qing Ye, Xixi Yang, Long Yu, Youchao Deng, Chengkun Wu, Tingjun Hou, and Dejun Jiang. A fused deep learning approach to transform novel drug repositioning. 2024.

[37] Alireza A.Tabatabaei, Mohammad Ebrahim Mahdavi, Ehsan Beiranvand, Sajjad Gharaghani, and Peyman Adibi. Graph neural network-based approaches to drug repurposing: A comprehensive survey. 2025.

[38] Javad Tayebi and Bagher BabaAli. Ekgdr: An end-to-end knowledge graph-based method for computational drug repurposing. 2024.

[39] P. Perdomo-Quinteiro, K. Wolstencroft, M. Roos, and Queralt-Rosinach. Knowledge graphs and explainable ai for drug repurposing on rare diseases.

[40] Ma Chunyu, Zhou Zhihan, Liu Han, and Koslicki David. Kgml-xdtd: a knowledge graph–based machine learning framework for drug treatment prediction and mechanism description. 2023.

[41] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. 2018.

[42] Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. Drugchat: Towards enabling chatgpt-like capabilities on drug molecule graphs. 2023.

[43] Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. Language models as knowledge embeddings. 2022.

[44] Daniel Domingo-Fernández, Yojana Gadiya, Abhishek Patel, Sarah Mubeen, Daniel Rivas-Barragan, Chris W. Diana, Biswapriya B. Misra, David Healey, Joe Rokicki, and Viswa Colluru. Causal reasoning over knowledge graphs leveraging drug-perturbed and disease-specific transcriptomic signatures for drug discovery. 2022.

[45] Yushan Liu, Marcel Hildebrandt, Mitchell Joblin, Martin Ringsquandl, Rime Raissouni, and Volker Tresp. Neural multi-hop reasoning with logical rules on biomedical knowledge graphs. 2021.

[46] Cassandra Königs, Marcel Friedrichs, and Theresa Dietrich. The heterogeneous pharmacological medical biochemical network pharmebinet. 2022.

[47] Jiannan Yang, Zhen Li, William Ka Kei Wu, Shi Yu, Qian Chu, and Qingpeng Zhang. Deep learning identifies explainable reasoning paths of mechanism of action for drug repurposing from multilayer biological network. 2022.

[48] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. 2019.

[49] Roberto Barile, Claudia d'Amato, and Nicola Fanizzi. Additive counterfactuals for explaining link predictions on knowledge graphs. 2025.

[50] Md Kamrul Islam, Diego Amaya-Ramirez, Bernard Maigret, Marie-Dominique Devignes, Sabeur Aridhi, and Malika Smaïl-Tabbone. Molecular-evaluated and explainable drug repurposing for covid-19 using ensemble knowledge graph embedding. 2023.

[51] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. 2020.

[52] Vijay Kumar Pounraja and Santhosh Girirajan. Oligogenic combinations of rare variants influence specific phenotypes in complex disorders, 2021.