DRKG Link Prediction with Graph Neural Networks

Technical Report

2026-02-19

Professor: Dr. Moosavi

Student 1: Sina Kamrava -- Student 1 ID: 40435105

Student 2: Amir Ahmadi Lari -- Student 2 ID: 40464266

This report summarizes the end-to-end pipeline, models, and outputs produced by the project notebooks. It covers data preprocessing, split construction, model training (GAT, R-GCN, R-GAT), evaluation, attention-based analysis, and query-time inference/recommendation.

1. Executive Summary

The project builds a compact multi-relational biomedical knowledge graph (KG) derived from DRKG and trains three graph neural network (GNN) approaches for link prediction: a GAT baseline, an R-GCN model, and an attention-based relational model (R-GAT).

On the internal held-out test split (38 positive edges with 50 sampled negatives per positive), R-GAT achieves the best ranking and classification performance (ROC-AUC 0.732, MRR 0.191, Hits@10 0.605), outperforming R-GCN and GAT. The pipeline also produces interpretability artifacts by aggregating attention weights per relation and by visualizing 2-hop subgraphs where edge width reflects learned attention.

2. Problem Statement

Given a biomedical KG with typed relations (entities such as compounds, genes, diseases, and anatomical terms), the goal is to predict missing or potential links between entities. This is framed as a supervised link prediction task where known edges are treated as positives and sampled non-edges as negatives. The project evaluates both (i) binary classification metrics over positive/negative samples and (ii) ranking metrics that measure whether true edges are ranked above negatives.

3. Data and Graph Construction

Two TSV datasets are used:

- **drkg_subgraph_120k.tsv:** A sampled DRKG subgraph used to build the training graph and internal splits.
- **drkg_test_holdout_20k.tsv:** An external holdout set used to test generalization (filtered to entities/relations seen in training).

After preprocessing, the graph statistics are:

Statistic	Value
Nodes (entities)	37,614
Edges (triples)	118,308
Relation types	107
Degree mean (in+out)	6.29
Degree median (in+out)	2
Degree 95th percentile	24
Degree max	373

Key processed artifacts saved under data/processed include entity/relation ID mappings and the typed edge list:

- entity2id.json / id2entity.json
- relation2id.json / id2relation.json
- graph_edges.pt (edge_index and edge_type)
- graph_meta.json (summary statistics)

3.1 Graph Topology and Relation Distribution

The node degree distribution is heavy-tailed: most nodes have small degree, while a few hub entities have substantially higher connectivity. Relation frequencies are also imbalanced, with a small number of relation types accounting for a large fraction of edges.

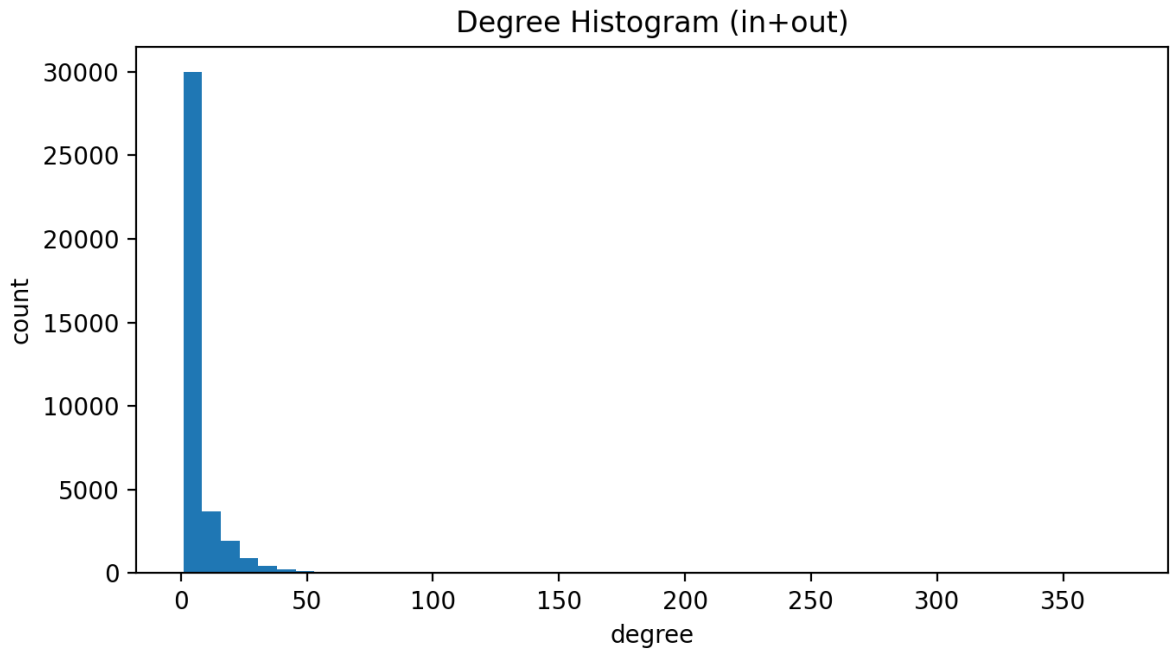

Figure 1. Degree histogram of the processed graph (in+out degree).

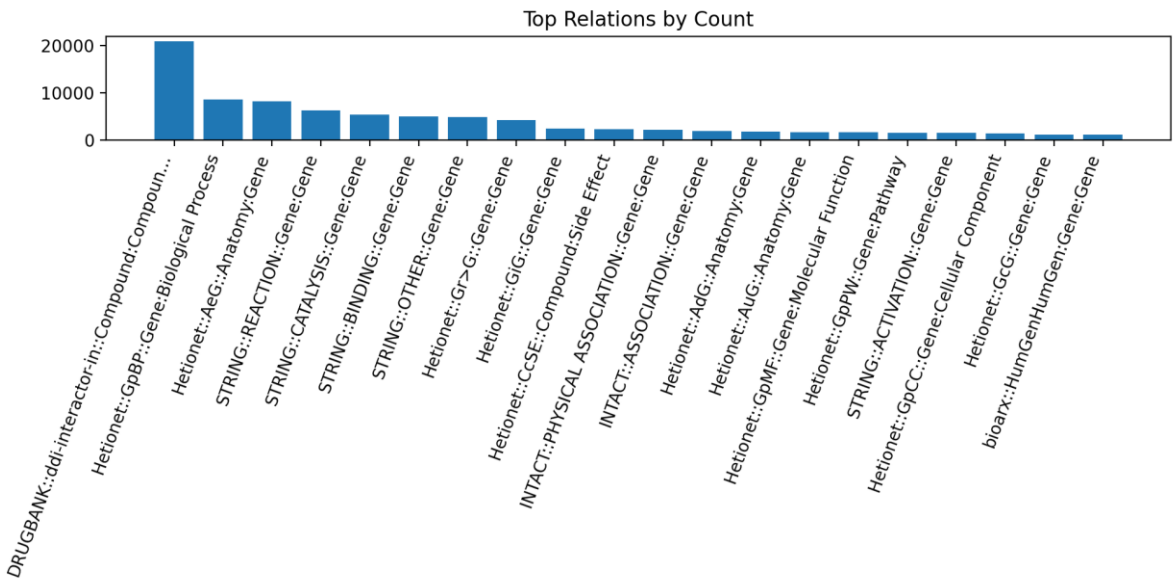

Figure 2. Top relation types by edge count in the processed graph.

Table 2 summarizes the top relation types by edge count (top-10).

Relation	Count
DRUGBANK::ddi-interactor-in::Compound:Compoun...	20,939
Hetionet::GpBP::Gene:Biological Process	8,672
Hetionet::AeG::Anatomy:Gene	8,177
STRING::REACTION::Gene:Gene	6,292
STRING::CATALYSIS::Gene:Gene	5,441
STRING::BINDING::Gene:Gene	5,027
STRING::OTHER::Gene:Gene	4,949
Hetionet::Gr>G::Gene:Gene	4,275
Hetionet::GiG::Gene:Gene	2,502
Hetionet::CcSE::Compound:Side Effect	2,379

4. Split Construction and Negative Sampling

The pipeline creates an internal train/validation/test split of target edges for evaluation. For validation and test, 50 negative edges are sampled per positive edge (K=50) to support ranking metrics.

Internal split sizes (positive edges):

Split	# Positive edges
Train	299
Validation	37
Test	38

Negative sampling configuration:

Split	Negatives per positive
Train (on-the-fly)	1
Validation	50
Test	50

Note on accuracy@0: because validation/test contain far more negatives than positives, a trivial always-negative classifier can achieve high accuracy. Therefore, ROC-AUC, PR-AUC, and ranking metrics (MRR/Hits@K) are emphasized.

5. Models

Three models are trained and compared:

1. **GAT (baseline):** Graph Attention Network over the observed graph structure; relation types are not explicitly modeled.
2. **R-GCN:** Relational Graph Convolutional Network; each edge carries a relation type and message passing is relation-aware.
3. **R-GAT:** Relational attention model that assigns attention weights to neighbor messages conditioned on relation types.

Model hyperparameters are configured in the training notebooks via a YAML config. The run used to generate the attention artifacts reports R-GAT settings of embedding dimension 32, 2 attention heads, and 8 bases (for parameter sharing across relations).

6. Training Procedure

Each model is trained for 100 epochs (as recorded in the training logs). The objective is link prediction using sampled negatives. Per-epoch logs report training loss and validation ROC-AUC/PR-AUC.

Table 4 summarizes validation performance and convergence behavior:

Model	Epochs	Best val ROC-AUC (epoch)	Best val PR-AUC (epoch)	Last loss	Last val ROC-AUC
GAT	100	0.627 (e100)	0.089 (e97)	0.186	0.627
R-GCN	100	0.720 (e39)	0.137 (e42)	0.029	0.647
R-GAT	100	0.812 (e98)	0.137 (e82)	0.270	0.812

R-GCN shows a classic overfitting pattern: the training loss continues decreasing while validation ROC-AUC peaks around epoch 39 and then declines. In contrast, R-GAT improves steadily toward the end.

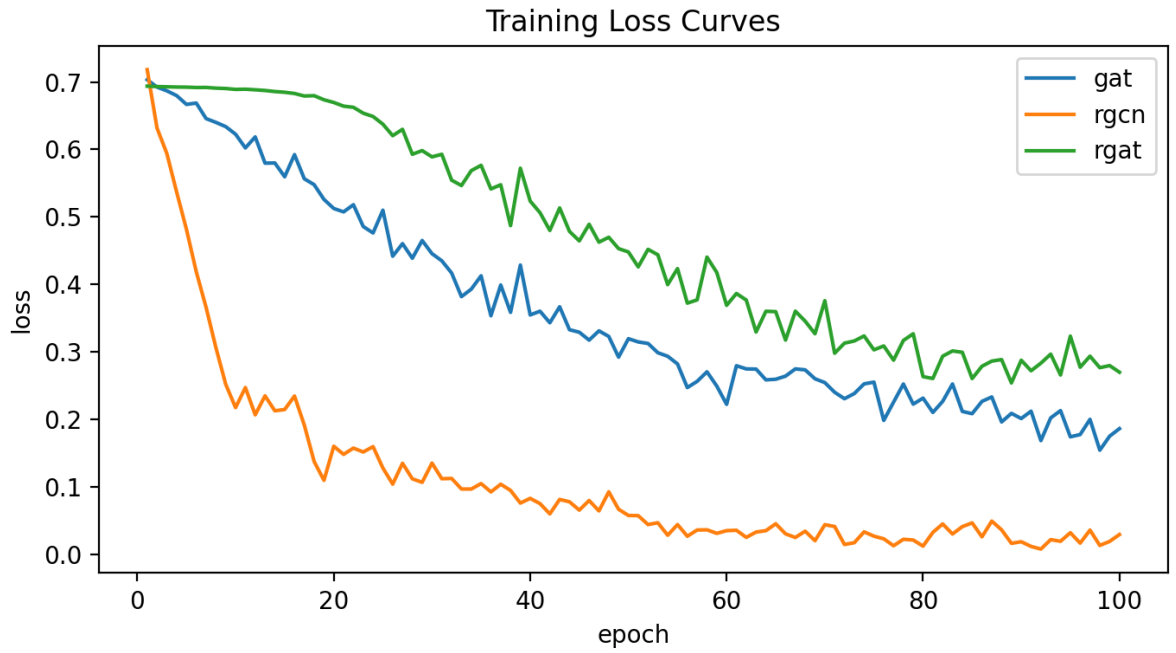

Figure 3. Training loss curves for GAT, R-GCN, and R-GAT.

7. Evaluation

Evaluation is performed on the internal test split using both binary and ranking metrics. Binary metrics (ROC-AUC and PR-AUC) are computed over the concatenated set of positives and sampled negatives. Ranking metrics are computed per positive by ranking it among its K sampled negatives (K=50).

Table 5 reports test performance (internal test split).

Model	ROC-AUC	MRR	Hits@10
gat	0.576	0.159	0.316
rgcn	0.640	0.179	0.421
rgat	0.732	0.191	0.605

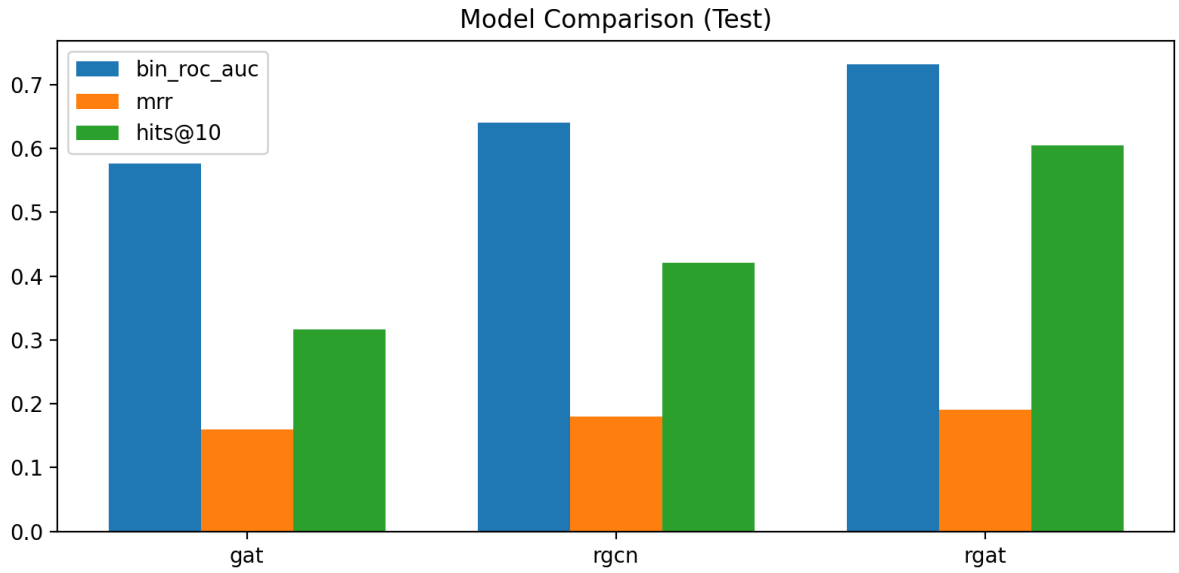

Figure 4. Model comparison on the internal test split (ROC-AUC, MRR, Hits@10).

R-GAT provides the strongest overall results. The largest improvement is observed in Hits@10, indicating substantially better top-k ranking quality compared to the baseline models.

8. Attention-Based Analysis (R-GAT)

To interpret R-GAT behavior, attention weights are aggregated by relation type. This highlights which relation families the model relies on most during message passing.

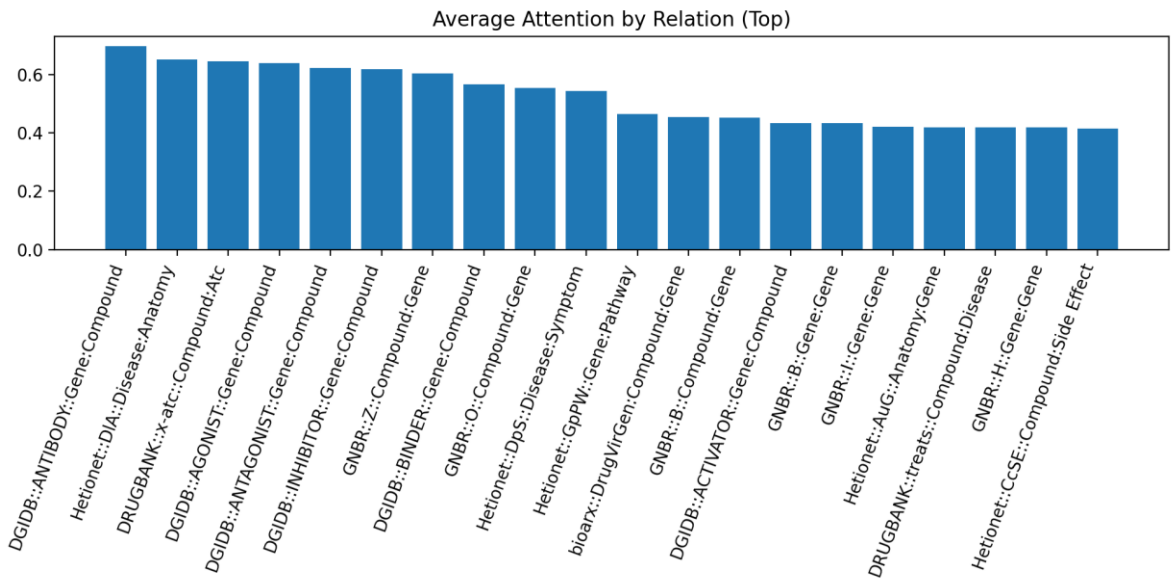

Figure 5. Average attention weight per relation type (top relations).

Table 6 lists the top-10 relations by average attention (higher implies greater model focus).

Relation	Avg. attention	Edge count
DGIDB::ANTIBODY::Gene:Compound	0.696	188
Hetionet::DlA::Disease:Anatomy	0.653	354
DRUGBANK::x-atc::Compound:Atc	0.646	536
DGIDB::AGONIST::Gene:Compound	0.640	345
DGIDB::ANTAGONIST::Gene:Compound	0.622	345
DGIDB::INHIBITOR::Gene:Compound	0.618	389
GNBR::Z::Compound:Gene	0.604	342
DGIDB::BINDER::Gene:Compound	0.567	143
GNBR::O::Compound:Gene	0.553	383
Hetionet::DpS::Disease:Symptom	0.543	350

Notably, relations with high attention are not necessarily the most frequent. This suggests the model learns to prioritize more informative relation types rather than simply reflecting relation frequency.

8.1 2-Hop Subgraph Explanations

The project also visualizes 2-hop neighborhoods around selected cases. In these plots, edge thickness is proportional to the learned attention, providing a qualitative explanation of which connections contribute most to the model's signal.

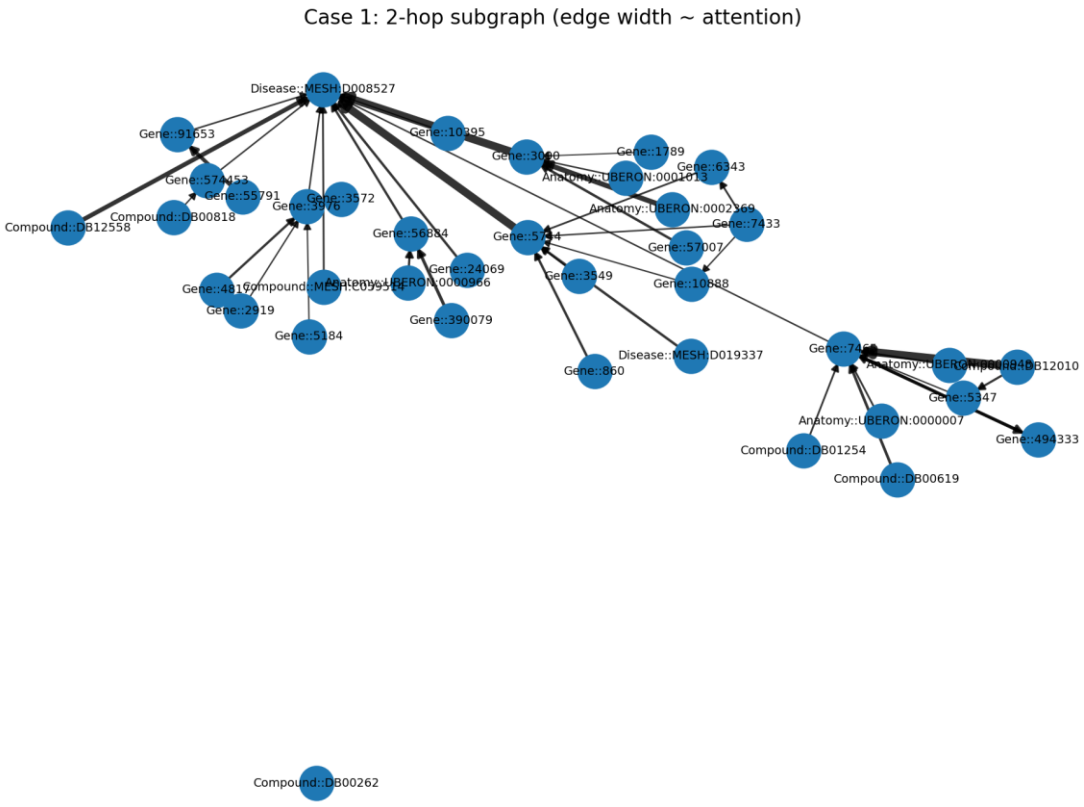

Figure 6. Example 2-hop subgraph visualization (edge width proportional to attention).

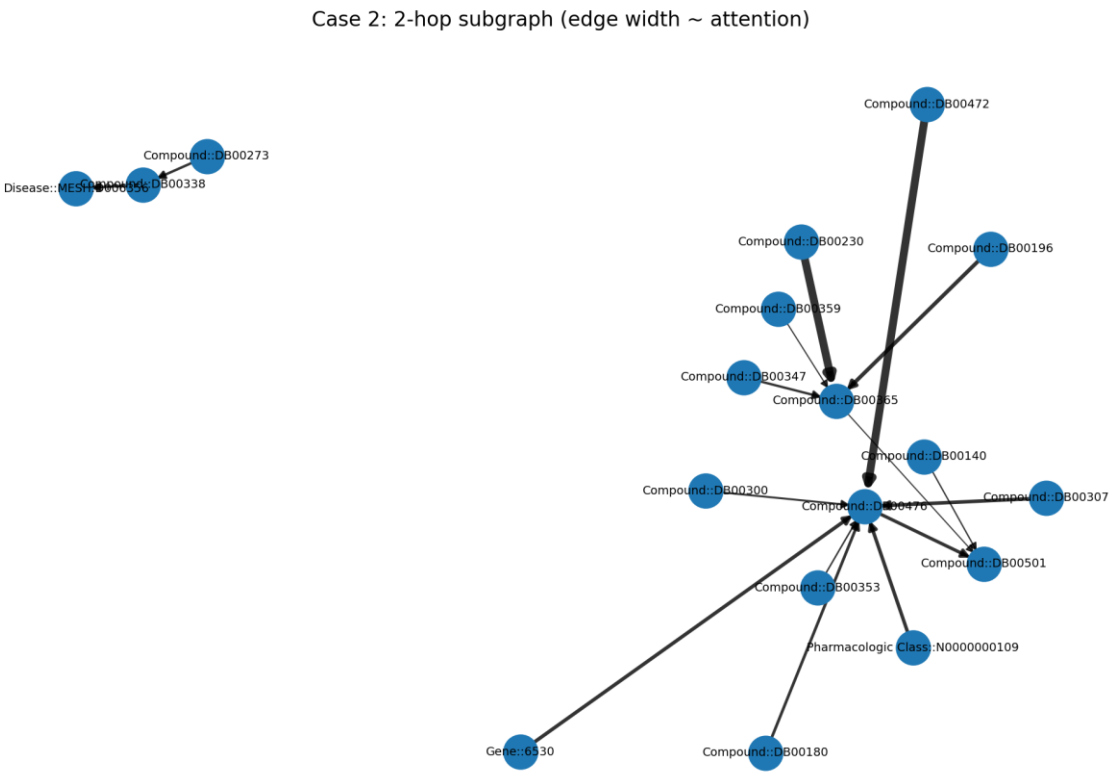

Figure 7. Example 2-hop subgraph visualization (edge width proportional to attention).

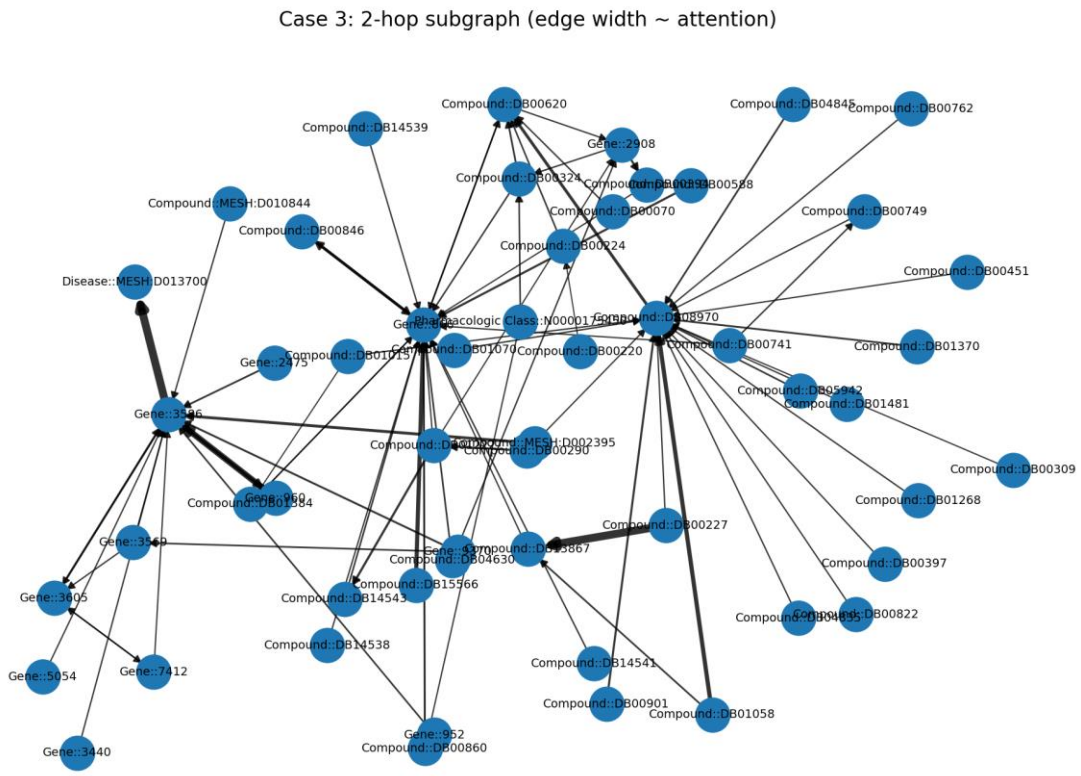

Figure 8. Example 2-hop subgraph visualization (edge width proportional to attention).

9. External Holdout Set

An additional test set (drkg_test_holdout_20k.tsv) is used for broader generalization checks. The holdout evaluation is configured to keep only triples whose entities and relations exist in the training graph (to avoid cold-start cases).

Holdout statistic	Value
Holdout triples	19,592
Require entities in training	True
Require relations in training	True
Eligible relations covered	95 / 95

Top relation types in the holdout set (top-5):

Relation	Count
DRUGBANK::ddi-interactor-	2,780

in::Compound:Compound	
Hetionet::AeG::Anatomy:Gene	1,058
Hetionet::GpBP::Gene:Biological Process	956
STRING::REACTION::Gene:Gene	880
STRING::CATALYSIS::Gene:Gene	769

10. Query-Time Inference and Recommendation

The pipeline supports querying a specific (drug, disease) pair and generating top-k compound recommendations for a given disease using the trained R-GAT model.

10.1 Pair Scoring Example

Input drug query: Compound::DB01234 (resolved to Compound::DB01234, id=3295).

Input disease query: Disease::MESH:D014774 (resolved to Disease::MESH:D017674, id=31631). Note that the resolver may choose the closest matching entity if an exact match is unavailable.

Model	Logit	Probability (sigmoid)
gat	-3.339	0.034264
rgcn	-18.567	0.000000
rgat	-7.607	0.000497

10.2 Top-K Recommendations (R-GAT)

The following table shows the top-10 predicted compounds for disease Disease::MESH:D017674 (id=31631). The probabilities are primarily used for ranking and may not be perfectly calibrated.

Compound	Score (prob_rgat)
Compound::DB14777	0.001301
Compound::DB13776	0.001301
Compound::DB01604	0.001233
Compound::DB14676	0.001177
Compound::DB12198	0.001177
Compound::CHEMBL2109511	0.001115

Compound::CHEMBL52030	0.001087
Compound::CHEMBL58757	0.001087
Compound::CHEMBL3545006	0.001068
Compound::DB12108	0.001068

11. Limitations and Future Work

Key limitations observed from the artifacts:

- The internal validation/test splits are small (37/38 positives), so estimates may have high variance.
- Negative sampling uses a fixed K=50 for ranking evaluation; different K or hard-negative strategies could change results.
- Relation and degree distributions are highly imbalanced (long-tail). Per-relation and per-degree evaluation would provide deeper insight.
- Accuracy@0 is not reliable under heavy class imbalance; ranking metrics and PR-AUC are more informative.

Potential next steps:

- Add early stopping/checkpoint selection using best validation ROC-AUC (especially important for R-GCN).
- Evaluate on the full external holdout set with consistent ranking metrics and confidence intervals.
- Report macro-averaged metrics across relation types and stratify performance by node degree.
- Improve entity resolution for query inference by requiring exact matches or returning uncertainty scores.

Appendix A. Generated Outputs (Inventory)

The project writes outputs into the following folders (relative to the project root):

- data/processed: processed graph tensors and ID mappings
- output/logs: per-epoch training logs for each model (CSV)
- output/metrics: evaluation metrics and attention summaries (CSV/JSON)
- output/figures: plots (degree histogram, relation counts, training curves, model comparison, attention plots)
- output/queries: pair scoring JSON and top-k recommendation CSVs
- output/subgraph: 2-hop subgraph visualizations with attention-weighted edges

For portability, it is recommended to keep all paths relative to the project root when running the notebooks. Some saved JSON summaries may contain absolute paths from the original execution environment; these do not affect results.