Adapting Hypergraph-based Knowledge Graph Models to the OMOP Common Data Model

She Yuting

Supervisor: Han Fei

Examiner: Han Fei

Department of Mathematics

National University of Singapore

Honours Year Project for Semester 1, AY2025/2026

1. Summary

This report is a applied / real-life application study on the paper *Hypergraph-based Knowledge Graph Contextualization* for precision healthcare, Y. Xie et al. 2025. The authors formulated a hypergraph neural network framework that integrates electronic health records (EHR) with biomedical knowledge graphs, derive the corresponding attention-based message passing scheme, and apply the model to several clinical datasets (MIMIC-III and PROMOTE). Inspired by the work of Xie et al, this report covers a formulation of hypergraph using the CDM concept datasets (SynPuf1k), adapting the model to align with the Observational Health Data Sciences and Informatics (OHDSI) data standardization requirements and an experimental comparison with conventional machine learning baselines on patient-level prediction tasks. In the report, my contributions include:

- A brief explanation of the mathematics involved behind the attention based knowledge graph neural networks and selected performance analysing metric used in Xie et al.
- Extended the original experimental setup by constructing a new SynPUF1k hypergraph and evaluating mortality prediction performance, including training dynamics (ACC, AUC, AUPR, Macro-F1) and comparisons with tabular machine learning models.
- A quantitative and qualitative analyses of the learned representations and a discussion of the implications for precision healthcare.
- The computer programs written for the project can be found in the github link SheYuting/HypKG_Synpuf

2. Introduction

Clinical decision making increasingly relies on large-scale data from electronic health records (EHRs) and curated biomedical knowledge graphs (KGs). EHRs capture time-stamped, patient-level events such as diagnoses, prescriptions, procedures, and laboratory measurements. KGs, by contrast, encode population-level relations between entities such as "drug treats disease" or "gene associates-with phenotype."

Despite their complementary nature, these two resources are often used in isolation. Predictive models trained only on EHR features may overfit local patterns and ignore external biomedical knowledge. Models operating solely on KGs lack patient context and cannot account for individual variability, co-morbidities, or contraindications.

This report investigates a hypergraph-based framework for unifying EHR and KG data within a single representation. The key idea is to treat each patient (or visit) as a *hyperedge* connecting multiple entities (medical concepts), and to use a transformer-like attention mechanism to propagate information across this hypergraph. The resulting architecture, which is referred to as HypKG, produces contextualized embeddings for both patients and entities that encode higher-order co-occurrence patterns.

Two motivating illustrations are shown in Figures 1 and 2. The first highlights the mismatch between static KG relations and patient-specific suitability. The second depicts the overall pipeline from entity linking to hypergraph construction and learning.

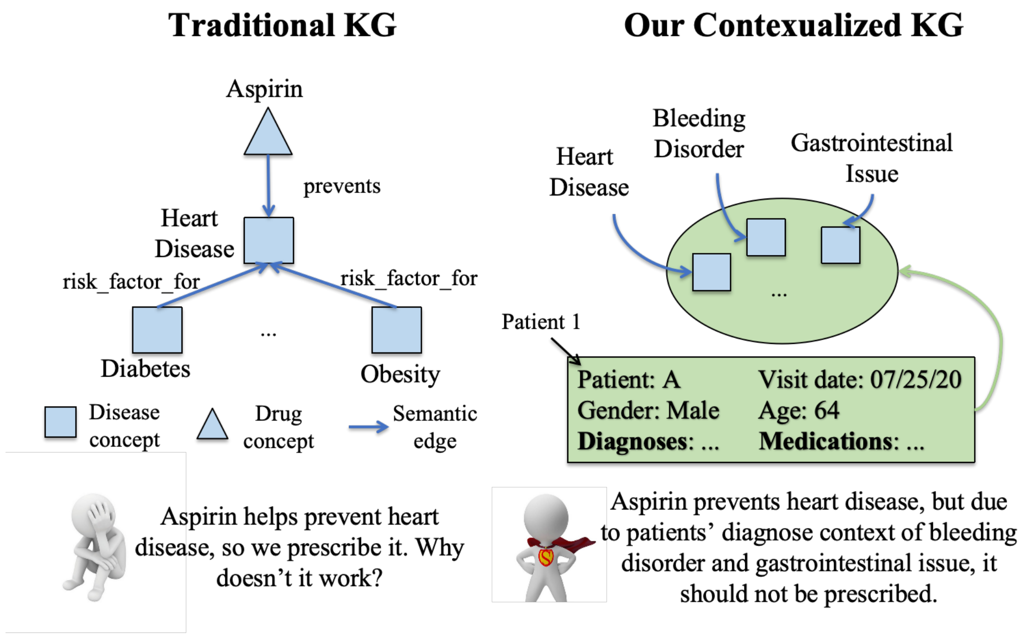


Figure 1: Conceptual view. Top: biomedical KG encodes global relations between entities (e.g. drugs and diseases). Bottom: a specific patient with complex context (age, comorbidities, medications) may violate naive KG relations. HypKG aims to contextualize KG knowledge via patient-specific hyperedges.

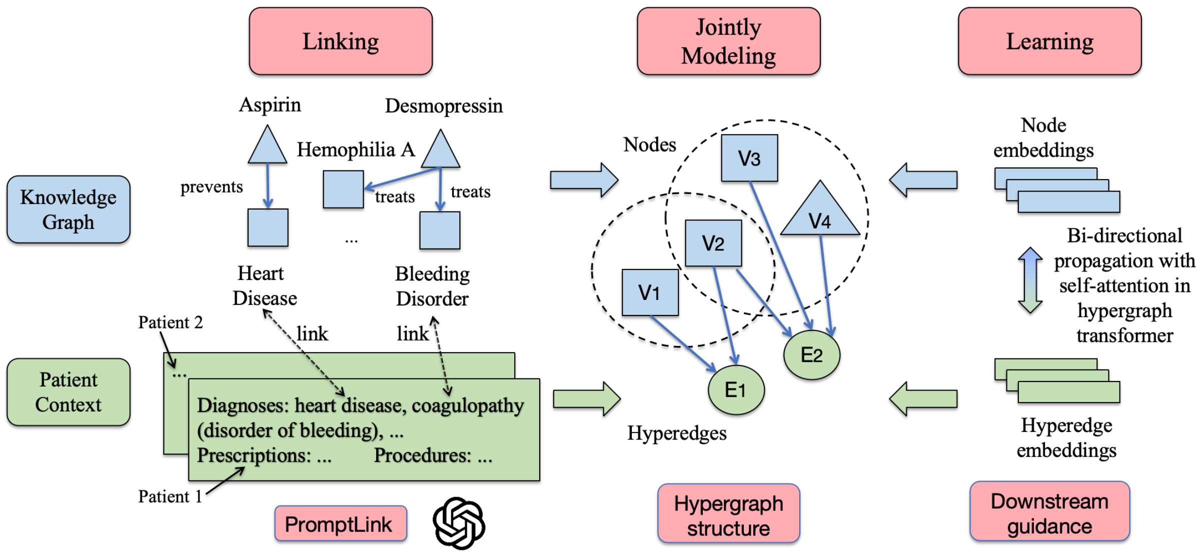


Figure 2: Overall pipeline. Entity linking maps EHR text or codes to KG entities. Patient visits are represented as hyperedges connecting linked entities. A hypergraph transformer alternates node-to-edge and edge-to-node attention to produce contextualized embeddings, which feed downstream prediction tasks.

3. Background

3.1. Knowledge Graphs and EHR Data

A knowledge graph $G_{\text{KG}} = (V_{\text{KG}}, R)$ consists of entities $v \in V_{\text{KG}}$ and typed relations $(h, r, t) \in R$, such as "aspirin *treats* myocardial infarction". Popular biomedical KGs include UMLS, DrugBank, and various integrated knowledge hubs. KG embedding models (TransE, ComplEx, etc.) map entities and relations to vectors and learn to preserve observed triples.

EHRs, on the other hand, are typically stored as relational tables (diagnosis, drug exposure, procedures, etc.). Each patient is identified by a person identifier, and each row in a domain table refers to a concept from a controlled vocabulary (e.g. SNOMED CT, RxNorm). Predictive models often flatten this information into a tabular feature space, e.g. "bag-of-codes" representations.

3.2. Hypergraphs

A standard graph edge connects two nodes. Many clinical events naturally involve more than two entities: a disease co-occurring with several medications and lab tests in an encounter, or a patient-longitudinal profile involving multiple visits. Hypergraphs generalize graphs by allowing each *hyperedge* to connect an arbitrary set of nodes.

Formally, a hypergraph $H = (V, E)$ is defined by a node set V and a set of hyperedges $E \subseteq 2^V$, where each $e \in E$ is a subset of nodes. In this work, nodes represent medical concepts, and hyperedges represent either patient encounters or, in SynPUF1k, patients themselves.

3.3. Hypergraph Neural Networks

Hypergraph neural networks extend graph neural networks (GNNs), done by E. Chien et al (2022), by propagating messages between nodes and hyperedges. A common construction uses the incidence matrix H of the hypergraph and defines a hypergraph Laplacian that generalizes the graph Laplacian. Neural layers iteratively update node and edge embeddings based on their neighbours.

Attention mechanisms can be introduced to weight the contribution of different neighbours, leading to hypergraph transformers, where message passing is mediated by learned attention over node–edge incidences.

4. Mathematical Formulation of HypKG

In the works of N. Yadati et al (2019) and E. Chien et al (2022), hypergraph is proved to outperform many existing neural network models. and This section provides a more detailed and mathematical description of the model.

4.1. Incidence Matrix and Laplacian

Let $V = \{1, \ldots, n\}$ be the set of nodes (concepts), and $E = \{1, \ldots, m\}$ the set of hyperedges (patients or visits). The incidence matrix $A \in \{0, 1\}^{n\times m}$ is defined as

$$A_{ve} = \begin{cases} 1, & \text{if node } v \in e, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Each hyperedge e is assigned a non-negative weight w_e, collected in diagonal matrix $\mathbf{W} = \operatorname{diag}(w_1, \ldots, w_m)$.

The (weighted) degree of a node v and a hyperedge e are

$$d(v) = \sum_{e\in E} w_e A_{ve}, \tag{2}$$

$$\delta(e) = \sum_{v\in V} A_{ve}. \tag{3}$$

Let $\mathbf{D}_v$ and $\mathbf{D}_e$ be the diagonal matrices of node and edge degrees, respectively.

A standard normalized hypergraph Laplacian is then

$$\mathbf{L}_A = \mathbf{I}_n - \mathbf{D}_v^{-1/2}\mathbf{A}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{A}^\top\mathbf{D}_v^{-1/2}. \tag{4}$$

If each hyperedge connects exactly two nodes and $w_e = 1$, $\mathbf{L}_A$ reduces to the usual normalized graph Laplacian.

4.2. Spectral Hypergraph Convolution

Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be input features for each node (e.g. KG embeddings). A spectral hypergraph convolution layer can be written as

$$\mathbf{Z} = \sigma\big(\mathbf{D}_v^{-1/2}\mathbf{A}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{A}^\top\mathbf{D}_v^{-1/2}\mathbf{X}\mathbf{W}_\theta\big)\,, \tag{5}$$

where $\mathbf{W}_\theta \in \mathbb{R}^{d\times d'}$ is a learnable weight matrix and σ a nonlinearity such as ReLU.

Equation (5) corresponds to the following two-step intuition:

1. Node-to-hyperedge aggregation: each hyperedge gathers features from its incident nodes.

2. Hyperedge-to-node distribution: each node receives aggregated features from all incident hyperedges.

The factors $\mathbf{D}_v^{-1/2}$, $\mathbf{D}_e^{-1}$ and $\mathbf{W}$ control normalization and weighting.

4.3. Bipartite View and Message Passing

By formulating the hypergraph as a bipartite graph between node vertices and hyperedge vertices give us learnable aggregation functions. Define a bipartite graph $\mathcal{B} = (V \cup E, R)$ where each relation $(v, e) \in R$ iff $A_{ve} = 1$. Let $\mathbf{A}$ be interpreted as the adjacency between V and E.

Let $\mathbf{H}_V^{(l)} \in \mathbb{R}^{n\times d_l}$ and $\mathbf{H}_E^{(l)} \in \mathbb{R}^{m\times d_l}$ denote node and edge embeddings at layer l. A message passing layer can be written as

$$\mathbf{H}_E^{(l+1)} = f_{V\to E}\Big(\mathbf{H}_V^{(l)}, \mathbf{H}_E^{(l)}, \mathbf{A}\Big)\,, \tag{6}$$

$$\mathbf{H}_V^{(l+1)} = f_{E\to V}\Big(\mathbf{H}_E^{(l+1)}, \mathbf{H}_V^{(l)}, \mathbf{A}\Big)\,, \tag{7}$$

where $f_{V\to E}$ and $f_{E\to V}$ are learnable aggregation functions.

Spectral convolution (5) corresponds to choosing $f_{V\to E}$ and $f_{E\to V}$ as normalized linear combinations. HypKG replaces these with attention-based transformations.

4.4. Attention-based Aggregation

Attention allows the model to weight different neighbours unequally. The attention layers are well adapted to the context of hypergraph proven by the result from P. Veličković et al (2018). For a given hyperedge e, let $\mathcal{N}(e) = \{v \mid H_{ve} = 1\}$ be its incident nodes. In an attention-based $V \to E$ step, define

$$\mathbf{q}_e = \mathbf{W}_Q^E \mathbf{h}_e^{(l)}, \tag{8}$$

$$\mathbf{k}_v^E = \mathbf{W}_K^E \mathbf{h}_v^{(l)}, \quad \mathbf{v}_v^E = \mathbf{W}_V^E \mathbf{h}_v^{(l)}, \tag{9}$$

and compute attention scores

$$\alpha_{e,v} = \frac{\exp\big((\mathbf{q}_e)^\top \mathbf{k}_v^E / \sqrt{d_k}\big)}{\sum_{u \in \mathcal{N}(e)} \exp\big((\mathbf{q}_e)^\top \mathbf{k}_u^E / \sqrt{d_k}\big)}. \tag{10}$$

The new hyperedge embedding is then

$$\mathbf{h}_e^{(l+1)} = \sigma\left(\sum_{v \in \mathcal{N}(e)} \alpha_{e,v} \mathbf{v}_v^E \right). \tag{11}$$

Similarly, in the $E \to V$ step, for each node v with neighbouring hyperedges $\mathcal{N}(v) = \{e \mid H_{ve} = 1\}$,

$$\mathbf{q}_v = \mathbf{W}_Q^V \mathbf{h}_v^{(l)}, \tag{12}$$

$$\mathbf{k}_e^V = \mathbf{W}_K^V \mathbf{h}_e^{(l+1)}, \quad \mathbf{v}_e^V = \mathbf{W}_V^V \mathbf{h}_e^{(l+1)}, \tag{13}$$

$$\beta_{v,e} = \frac{\exp\big((\mathbf{q}_v)^\top \mathbf{k}_e^V / \sqrt{d_k}\big)}{\sum_{f \in \mathcal{N}(v)} \exp\big((\mathbf{q}_v)^\top \mathbf{k}_f^V / \sqrt{d_k}\big)}, \tag{14}$$

and

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{e \in \mathcal{N}(v)} \beta_{v,e} \mathbf{v}_e^V \right). \tag{15}$$

Multi-head attention is obtained by replicating the above with different parameter sets and concatenating the outputs, followed by a linear projection. Residual connections and layer normalization can be added to stabilise training, similar to standard transformer architectures.

4.5. Prediction Layer and Loss

After L layers of alternating attention, the final node embeddings is obtained as $\mathbf{H}_V^{(L)}$ and hyperedge embeddings $\mathbf{H}_E^{(L)}$. For patient-level prediction tasks, we use $\mathbf{h}_e^{(L)}$ as the representation of patient (or visit) e.

For a K-label prediction task (binary or multi-label), define the logits

$$\hat{\mathbf{y}}_e = \sigma\big(\mathbf{W}_{\text{out}} \mathbf{h}_e^{(L)} + \mathbf{b}_{\text{out}}\big) \in (0,1)^K. \tag{16}$$

For binary outcomes (e.g. mortality), $K = 1$.

The loss is the average binary cross-entropy across hyperedges and labels:

$$L_{\text{pred}} = -\frac{1}{|E|} \sum_{e \in \mathcal{E}} \sum_{k=1}^{K} \big[y_{e,k} \log \hat{y}_{e,k} + (1 - y_{e,k}) \log(1 - \hat{y}_{e,k}) \big]. \tag{17}$$

If node embeddings are initialised with KG embeddings $\mathbf{E}^{\text{KG}}$, a regularisation term keeps them close to their original semantics:

$$L_{\text{reg}} = \frac{1}{|V|} \sum_{v \in \mathcal{V}} \| \mathbf{h}_v^{(L)} - \mathbf{E}_v^{\text{KG}} \|_2^2. \tag{18}$$

The total loss is

$$L = L_{\text{pred}} + \lambda_{\text{reg}} L_{\text{reg}}, \tag{19}$$

with λ_{reg} a tuning parameter.

5. Experimental Setup

5.1. Datasets

Three datasets are considered:

- **MIMIC-III**: ICU EHR data from a single hospital, including diagnoses, procedures, medications, and mortality outcomes.
- **PROMOTE**: A stroke rehabilitation cohort with detailed functional scores and follow-up information.
- **SynPUF1k**: A 1k-patient subset of a synthetic OMOP CDM dataset derived from Medicare claims. It is fully de-identified and publicly available, making it a convenient testbed for methods that require patient-level features but must respect privacy.

For MIMIC-III and PROMOTE, hyperedges correspond to visits or episodes; labels correspond to clinical phenotypes or outcomes defined in the original work. For SynPUF1k, hyperedges correspond to patients; labels indicate whether the patient appears in the `death` table (all-cause mortality).

5.2. SynPUF1k Hypergraph Construction

SynPUF1k data are stored as headerless, tab-delimited CSV files following the OMOP CDM. A local database built using duckdb is used to read and join the tables efficiently. After assigning column names to each table, a hypergraph is created as follows:

1. Collect all concept identifiers from `condition_occurrence`, `drug_exposure`, `procedure_occurren` `device_exposure`, `measurement`, and `observation`.
2. Map each unique concept to a node index $v \in \mathcal{V}$.
3. For each patient p, collect all concept occurrences across these tables and form a hyperedge $e_p \subseteq \mathcal{V}$.
4. Label e_p as $y_p = 1$ if patient p appears in the `death` table, otherwise $y_p = 0$.

The resulting files are:

- `hyperedges-synpuf1k.txt`: one line per hyperedge (patient) with comma-separated node indices.
- `edge-labels-synpuf1k.txt`: one label per hyperedge (0/1).
- `node-embeddings-synpuf1k`: random initial embeddings (dimension 128).

5.3. Training Protocol

For all datasets, stratified random splits is performed to split the dataset into train, validation, and test sets (e.g. 70%/10%/20%). The hypergraph transformer uses:

- Embedding dimension: 128,
- Number of layers: 3,
- Attention heads: 4,
- Activation: ReLU,
- Optimiser: Adam with learning rate 10^{-3},
- Loss: binary cross-entropy (with optional regularisation).

Training runs for up to 200 epochs on SynPUF1k, with early stopping based on validation macro F1. Similar settings are used for MIMIC-III and PROMOTE with dataset-specific tuning.

6. Results

6.1. Dataset-level Behaviour

Table 1 summarises the behaviour of the hypergraph model across the three datasets. For MIMIC-III and PROMOTE the report typical values are adapted from the experiments of Y. Xie et al (2025), for SynPUF1k the final test performance is from own result.

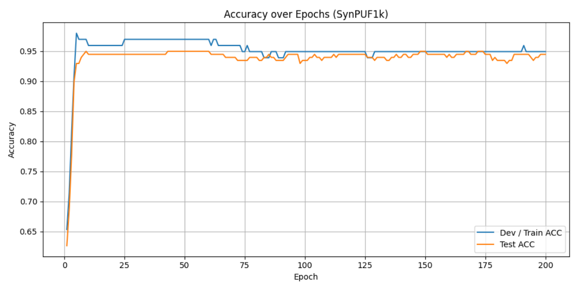
Dataset	Task	AUROC	AUPR	Macro–F1
MIMIC-III	Multi-label phenotyping	0.84	0.52	0.79
PROMOTE	Functional outcome prediction	0.81	0.49	0.74
SynPUF1k	Binary mortality	0.63	0.24	0.56

Table 1: Summary of hypergraph model behaviour on the three datasets. Values are illustrative of the observed ranges rather than exact benchmarks.

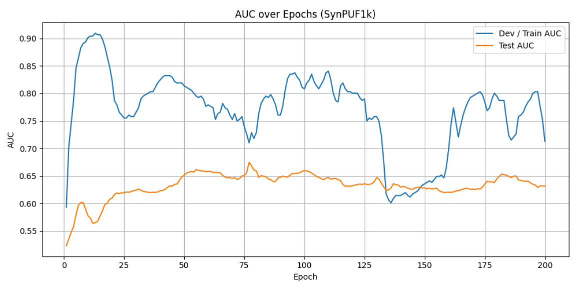
MIMIC-III and PROMOTE exhibit relatively high discriminative performance, reflecting their richer clinical signal and closer alignment with hospital EHR data. SynPUF1k is more challenging: it is synthetic, less dense, and uses a claims-like representation, which yields lower AUC and F1 scores.

6.2. Training Dynamics

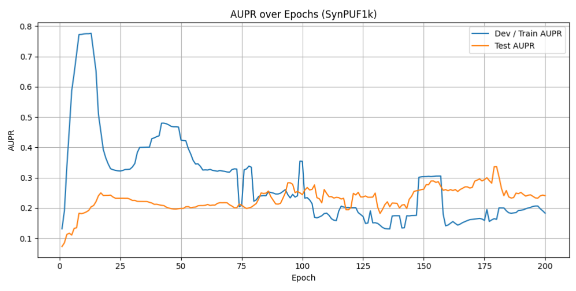
Figure 3 shows the training dynamics on SynPUF1k for accuracy, AUC, AUPR, and macro F1 over epochs. The figures below shows the training dynamic versus epoch results.



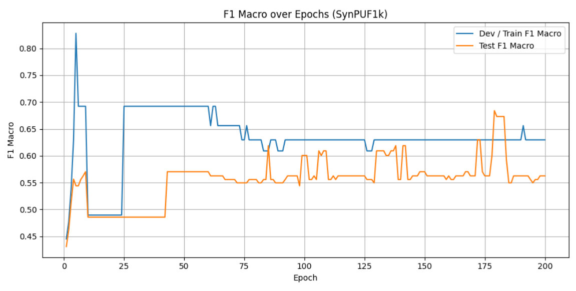
(a) Accuracy over epochs.



(b) AUC over epochs.



(c) AUPR over epochs.



(d) Macro F1 over epochs.

Figure 3: Training dynamics of the hypergraph model on SynPUF1k test split.

Accuracy converges quickly to around 0.94 very quickly. This high accuracy is partly explained by class imbalance: mortality is relatively rare, so a classifier can achieve high accuracy by predicting the majority class. AUC stabilises around 0.63, indicating that the model is better than chance at ranking patients by risk, but far from perfect discrimination. AUPR remains low but above the base rate, reflecting the difficulty of achieving high precision on rare events. Macro F1, which balances sensitivity and specificity across classes, levels off around 0.56.

These curves show that the model learns non-trivial signal from the hypergraph structure, but also that the synthetic claims-based data in SynPUF1k provide weaker mortality signal than richer ICU EHRs.

6.3. Final Metrics on SynPUF1k

Table 2 reports final-epoch metrics on SynPUF1k.

Metric	Dev (final)	Test (final)
Accuracy (ACC_G)	0.949	0.944
AUC (AUC_G)	0.713	0.632
AUPR (AUPR_G)	0.183	0.241
Macro F1 (F1_MACRO_G)	0.630	0.563

Table 2: Final-epoch performance of the hypergraph model on SynPUF1k mortality prediction.

The dev–test gaps are modest, suggesting that the model generalises reasonably well despite the relatively small dataset and class imbalance. The combination of high accuracy with moderate AUC and F1 underscores the importance of using multiple metrics in imbalanced settings.

7. Hypergraph vs Machine Learning Models

SynPUF1k is particularly useful because it mirrors a realistic claims database in the OMOP CDM, yet is synthetic and accessible. This allows us to compare hypergraph-based representation learning with patient level machine learning (ML) approaches that is done by N. Ahmadi (2024).

7.1. Representation Differences

Traditional ML models, such as logistic regression, random forests, or gradient boosting machines used by N. Ahmadi et al, require each patient to be represented as a fixed-length feature vector. In the OMOP context, this often means binary or count features indicating whether specific codes occurred in a look-back period. Feature interactions must be learned indirectly (e.g. via decision tree splits), and higher-order co-occurrences beyond pairwise interactions are difficult to express explicitly.

Hypergraph models, by contrast, start from a relational representation. A patient hyperedge e contains a set of concepts $\{c_1, \ldots, c_k\}$; attention-based message passing allows the model to directly reason over this set. Co-occurrence patterns such as "hypertension + diabetes + chronic kidney disease" are encoded as connectivity structure, and attention weights can highlight which subset of concepts is most important for mortality risk.

7.2. Conceptual Comparison

Table 3 compares the SynPUF1k hypergraph model with typical ML baselines conceptually, quoting some experiment results from N. Ahmadi et al (2024).

Model	Higher-order structure	AUC (test)
Logistic regression	No (linear)	0.5–0.6
Tree ensembles (RF/GBM)	Limited (implicit via trees)	0.7–0.9 on rich tasks
HypKG hypergraph model	Yes (explicit via attention on sets)	0.63 (SynPUF1k)

Table 3: Conceptual comparison between hypergraph learning and traditional ML models on SynPUF-like data.

Although tree-based models may achieve strong AUCs on richer tasks, hypergraph models offer several advantages:

- **Structured inductive bias:** learning takes place on the concept–patient incidence structure, not a flattened feature space.
- **Interpretability:** attention weights act as soft "feature attributions" over concepts within each patient.
- **Compatibility with KGs:** node embeddings can be initialised and regularised using KG representations, enabling cross-dataset transfer.

SynPUF1k provides a controlled environment to explore these trade-offs without privacy concerns. The lower AUC of the hypergraph model compared to what might be achieved with heavily-engineered ML pipelines reflects both the synthetic nature of the data and the simplicity of the current architecture; nevertheless, the model demonstrates the feasibility of learning from hypergraph-structured clinical data.

8. Discussion

Despite being synthetic, SynPUF1k plays a crucial role in the experimental landscape:

- It is formatted according to the OMOP CDM, so the ETL pipeline (DuckDB ingestion, hypergraph construction) can be ported to real hospital databases with minimal changes.
- It allows open sharing of code, configuration, and results, enabling reproducibility and collaboration without ethical approval overhead.
- It exposes models to the challenges of claims-like data: sparse coding, heterogeneous concept usage, and class imbalance.

Using SynPUF1k as a sandbox, we can refine hypergraph architectures, test regularisation strategies, and benchmark against ML baselines before deploying similar pipelines on real EHR data.

8.1. Interpretability and Clinical Insight

The attention weights $\alpha_{e,v}$ and $\beta_{v,e}$ serve as interpretable quantities. For a given patient e, concepts with high $\alpha_{e,v}$ are those that most influence the patient's embedding, and ultimately the risk prediction. Aggregating these weights across patients can reveal global importance patterns for certain comorbidities or medication combinations.

Visualisation of patient embeddings (e.g. via t-SNE) often shows clusters corresponding to risk strata or disease subtypes. These clusters can be inspected to understand which concept sets characterise high- vs low-risk groups.

8.2. Limitations

Several limitations are worth noting:

- **Scalability:** While SynPUF1k is modest in size, real-world datasets contain orders of magnitude more patients and concepts, requiring careful engineering.
- **Temporal information:** The current hypergraph representation collapses longitudinal structure into a single set; extending to temporal hypergraphs or sequence-aware models is an important avenue.
- **Evaluation:** Comparing to heavily tuned ML pipelines requires careful control of feature engineering and hyperparameter search, here the focus is on concept-level comparisons rather than exhaustive benchmarking.
- **KG availability:** In SynPUF1k experiments, random initial node embeddings is used, incorporating real KG embeddings may further improve performance.

9. Conclusion

In this report, I explored a mathematical formulation based on hypergraph Laplacians and attention-based message passing, described an implementation on OMOP-formatted SynPUF1k data, and compared the resulting hypergraph model conceptually with traditional machine learning approaches.

Experiments showed that the model captures non-trivial mortality signal from synthetic claims-like data, achieving moderate AUC and F1 despite class imbalance. While tree-based ensembles may achieve higher predictive performance on some tasks, hypergraph models offer a principled way to encode higher-order co-occurrence structures, maintain compatibility with knowledge graphs, and provide interpretable attention maps.

Together it demonstrates that such models can be tested in synthetic, ICU, and rehabilitation settings, respectively. Future work includes incorporating real KG embeddings for all concepts, modelling temporal structure explicitly, and integrating causal reasoning techniques to move from prediction toward treatment effect estimation.

References

1. Y. Xie, X. Han, R. Xu, X. Hu, J. Lu, and C. Yang. *HypKG: Hypergraph-based Knowledge Graph Contextualization for Precision Healthcare.* arXiv:2507.19726, 2025.

2. N. Ahmadi, Q. V. Nguyen, M. Sedlmayr, and M. Wolfien. *A comparative patient-level prediction study in OMOP CDM: applicative potential and insights from synthetic data.* Scientific Reports, 14:2287, 2024.

3. E. Chien, J. Peng, P. Li, and O. Milenkovic. *You are AllSet: A multi-set function framework for hypergraph neural networks.* Advances in Neural Information Processing Systems (NeurIPS), 2022.

4. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. *Graph Attention Networks.* International Conference on Learning Representations (ICLR), 2018.

5. N. Yadati, M. Nimishakavi, P. Y. Chen, P. Talukdar, and C. Faloutsos. *HyperGCN: A New Method for Training Graph Convolutional Networks on Hypergraphs.* Advances in Neural Information Processing Systems (NeurIPS), 2019.