GraphCare: Enhancing Healthcare Predictions With Personalized Knowledge Graphs

Pengcheng Jiang, Cao Xiao, Adam Cross, Jimeng Sun

(ICLR 2024)

Overview

- Motivation
- Challenges
- Method
- Experiments

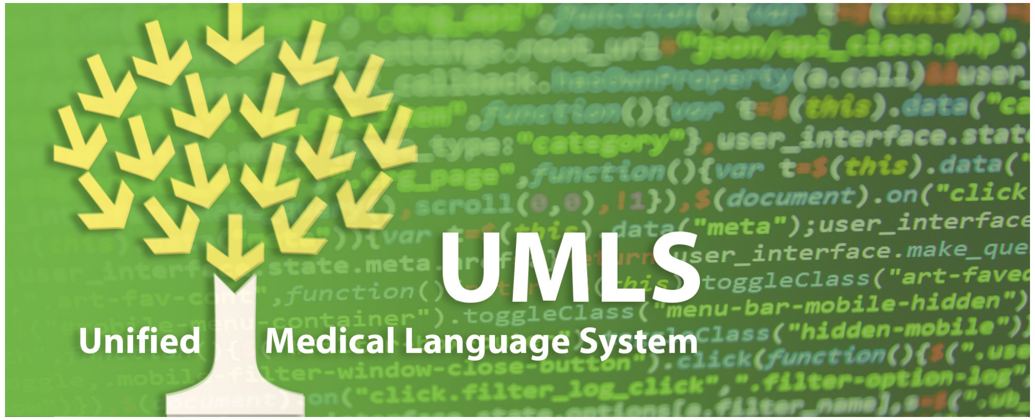
Motivation

1. Personalized KG is needed for personalized treatment.

2. Existing works mainly focus on
   1) simple hierarchical relations
   2) the inner graphical structure of EHR
   while there are many external biomedical KGs available

LLM as Knowledge Base

UMLS-KG



Challenges

1. How to construct personalized KGs utilizing external knowledge bases?

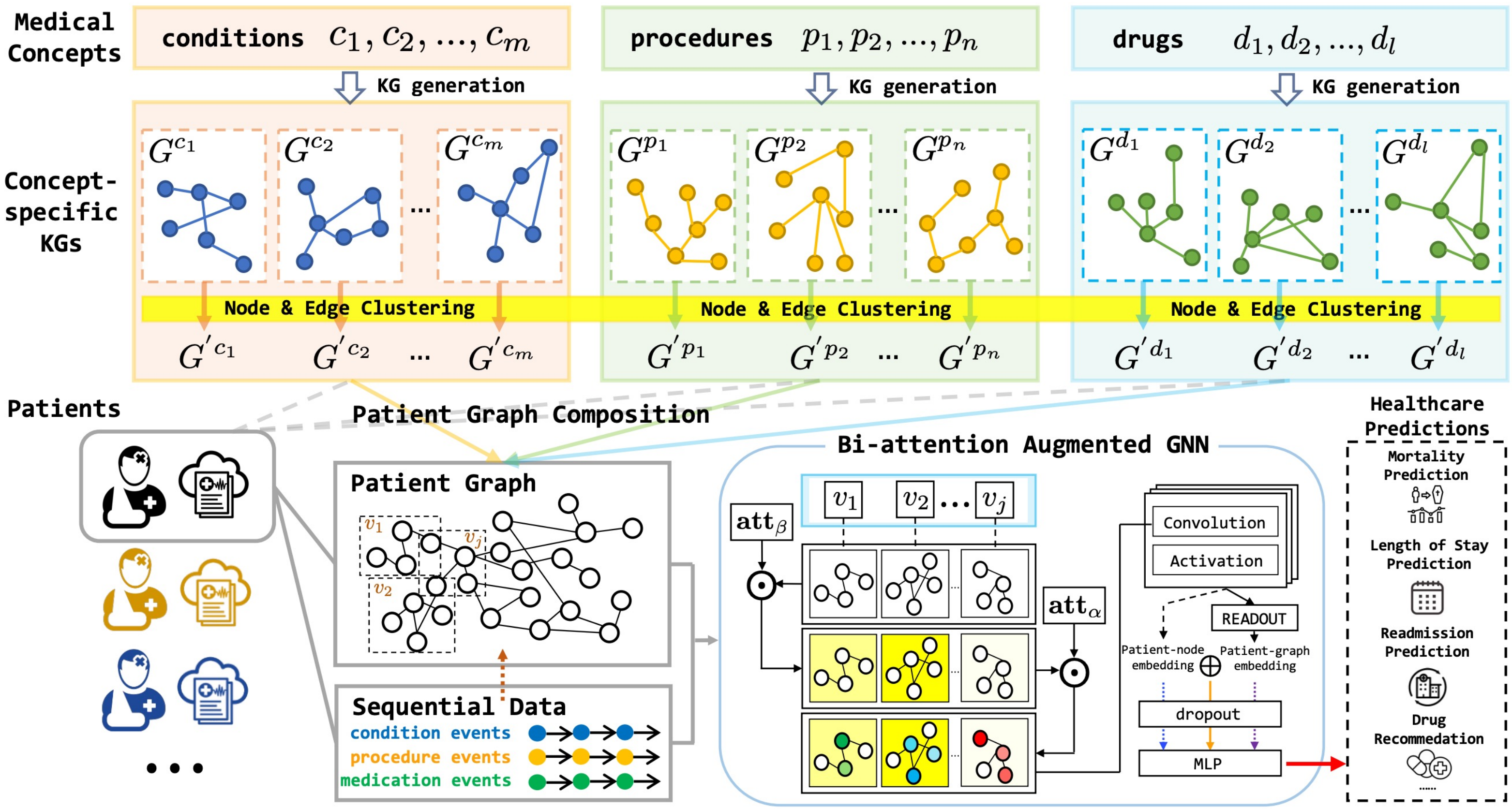
   - We construct medical concept-specific KGs

2. How to improve time-series clinical (EHR) predictions with those KGs?

   - We treat personalized knowledge graphs as patient representations

Method

Our proposed method - GraphCare

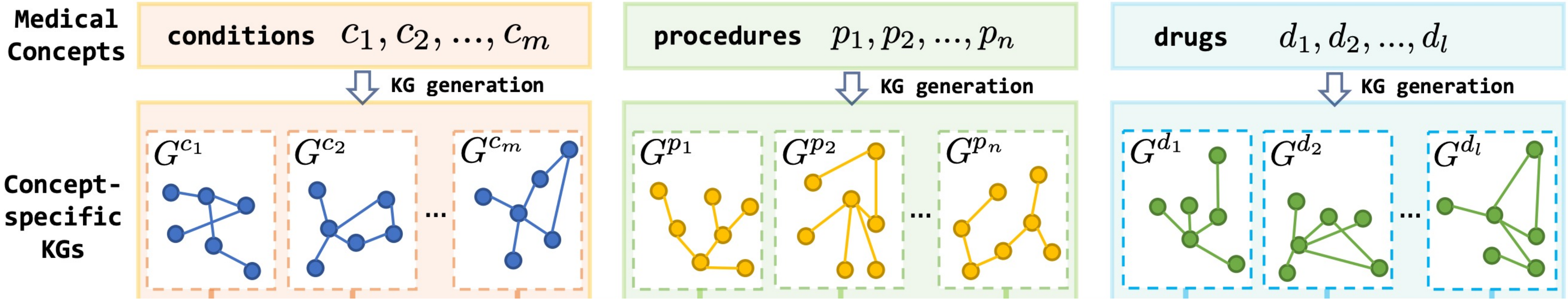


Step 1: Generate concept-specific KGs for every medical concept using LLM prompts and by subsampling from existing KGs. Perform clustering on nodes and edges across these KGs.
Step 2: For each patient, merge relevant concept-specific KGs to form a personalized KG.
Step 3: Employ Bi-attention Augmented (BAT) Graph Neural Network (GNN) to predict based on the personalized KGs.

Method

GraphCare – Concept-specific KG Generation

Step 1: Generate concept-specific KGs for every medical concept using LLM prompts and by subsampling from existing KGs. Perform clustering on nodes and edges across these KGs.



1.1 Knowledge extraction from LLM through prompting:

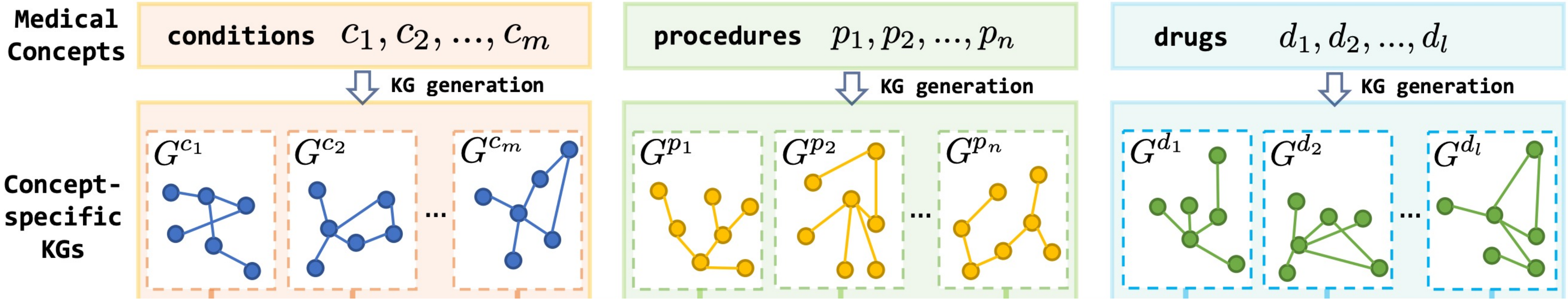
```
Given a prompt (a medical condition/procedure/drug), extrapolate as many relationships
as possible of it and provide a list of updates.
The relationships should be helpful for healthcare prediction (e.g., drug
recommendation, mortality prediction, readmission prediction …)
Each update should be exactly in format of [ENTITY 1, RELATIONSHIP, ENTITY 2]. The
relationship is directed, so the order matters.
Both ENTITY 1 and ENTITY 2 should be noun.
Any element in [ENTITY 1, RELATIONSHIP, ENTITY 2] should be conclusive, make it as
short as possible.
Do this in both breadth and depth. Expand [ENTITY 1, RELATIONSHIP, ENTITY 2] until the
size reaches 100.

{example}

prompt: {term}
updates:
```

```
if category == "condition":
example = \
"""Example:
prompt: systemic lupus erythematosus
updates: [[systemic lupus erythematosus, is an, autoimmune condition], [systemic
lupus erythematosus, may cause, nephritis], [anti-nuclear antigen, is a test for,
systemic lupus erythematosus], [systemic lupus erythematosus, is treated with,
steroids], [methylprednisolone, is a, steroid]]
"""
elif category == "procedure":
example = \
"""Example:
prompt: endoscopy
updates: [[endoscopy, is a, medical procedure], [endoscopy, used for, diagnosis],
[endoscopic biopsy, is a type of, endoscopy], [endoscopic biopsy, can detect,
ulcers]]
"""
elif category == "drug":
example = \
"""Example:
prompt: iobenzamic acid
updates: [[iobenzamic acid, is a, drug], [iobenzamic acid, may have, side
effects], [side effects, can include, nausea], [iobenzamic acid, used as, X-ray
contrast agent], [iobenzamic acid, formula, C16H13I3N2O3]]
"""
```

Method

GraphCare – Concept-specific KG Generation

Step 1: Generate concept-specific KGs for every medical concept using LLM prompts and by subsampling from existing KGs. Perform clustering on nodes and edges across these KGs.



1.2 Subgraph sampling from existing KG:

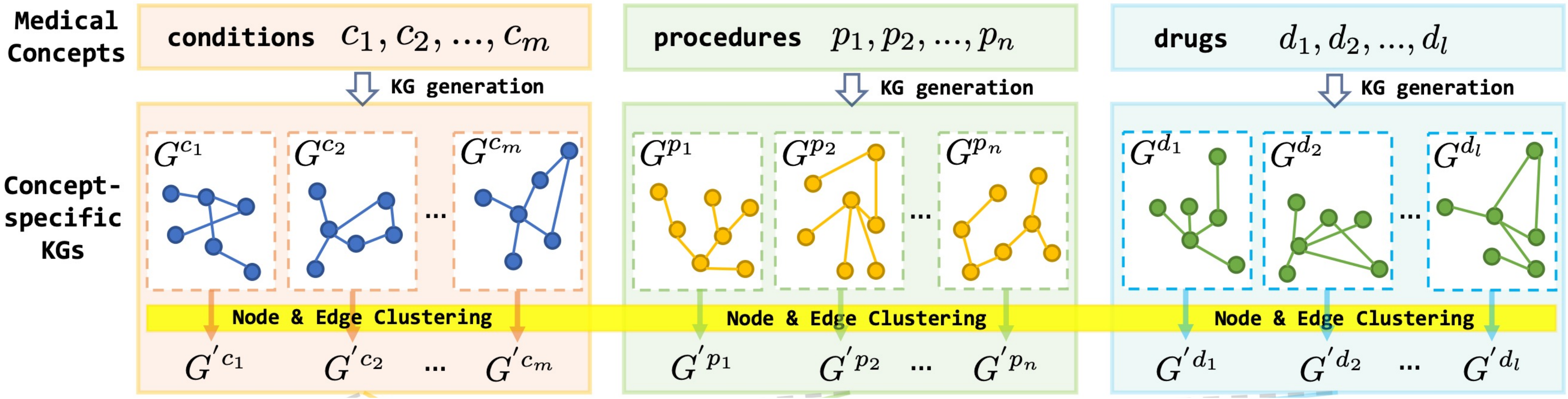
procedure SUBGRAPHSAMPLING(medical concept e, KG $\mathcal{G}$, hop limit κ, window size ϵ)
Initialize an empty list Q and an empty graph $G^e_{\mathrm{sub}(\kappa)} = (\mathcal{V}^e_{\mathrm{sub}(\kappa)}, \mathcal{E}^e_{\mathrm{sub}(\kappa)})$
Add e to Q
$\mathcal{V}^e_{\mathrm{sub}(\kappa)} \leftarrow \mathcal{V}^e_{\mathrm{sub}(\kappa)} \cup \{e\}$
for $i = 1$ to κ **do**
Initialize an empty list Q_{next}
for all ent $\in Q$ **do**
if $i = 1$ **then**
Retrieve all triples $(\mathrm{ent}, \mathrm{rel}, \mathrm{ent}')$ or $(\mathrm{ent}', \mathrm{rel}, \mathrm{ent})$ from $\mathcal{G}$
else
Randomly retrieve ϵ triples $(\mathrm{ent}, \mathrm{rel}, \mathrm{ent}')$ or $(\mathrm{ent}', \mathrm{rel}, \mathrm{ent})$ from $\mathcal{G}$
end if
Add retrieved triples to $\mathcal{E}^e_{\mathrm{sub}(\kappa)}$
$\mathcal{V}^e_{\mathrm{sub}(\kappa)} \leftarrow \mathcal{V}^e_{\mathrm{sub}(\kappa)} \cup \{\mathrm{ent}'\}$
Add ent$'$ to Q_{next}
end for
$Q \leftarrow Q_{\mathrm{next}}$
end for
return $G^e_{\mathrm{sub}(\kappa)}$

(For the first hop, we retrieve all the triples containing the concept. For the second (and higher) hop, we randomly retrieve triples of window size ϵ)

Method

GraphCare – Concept-specific KG Generation

Step 1: Generate concept-specific KGs for every medical concept using LLM prompts and by subsampling from existing KGs. Perform clustering on nodes and edges across these KGs.



1.3 Node & Edge Clustering

Based on the word embedding of nodes and edges, we apply agglomerative clustering to get two mappings:

$$\mathcal{C}_{\mathcal{V}} : \mathcal{V} \rightarrow \mathcal{V}'$$
$$\mathcal{C}_{\mathcal{E}} : \mathcal{E} \rightarrow \mathcal{E}'$$

New global graph (node clusters as new nodes; edge clusters as new edges):

$$G' = (\mathcal{V}', \mathcal{E}')$$

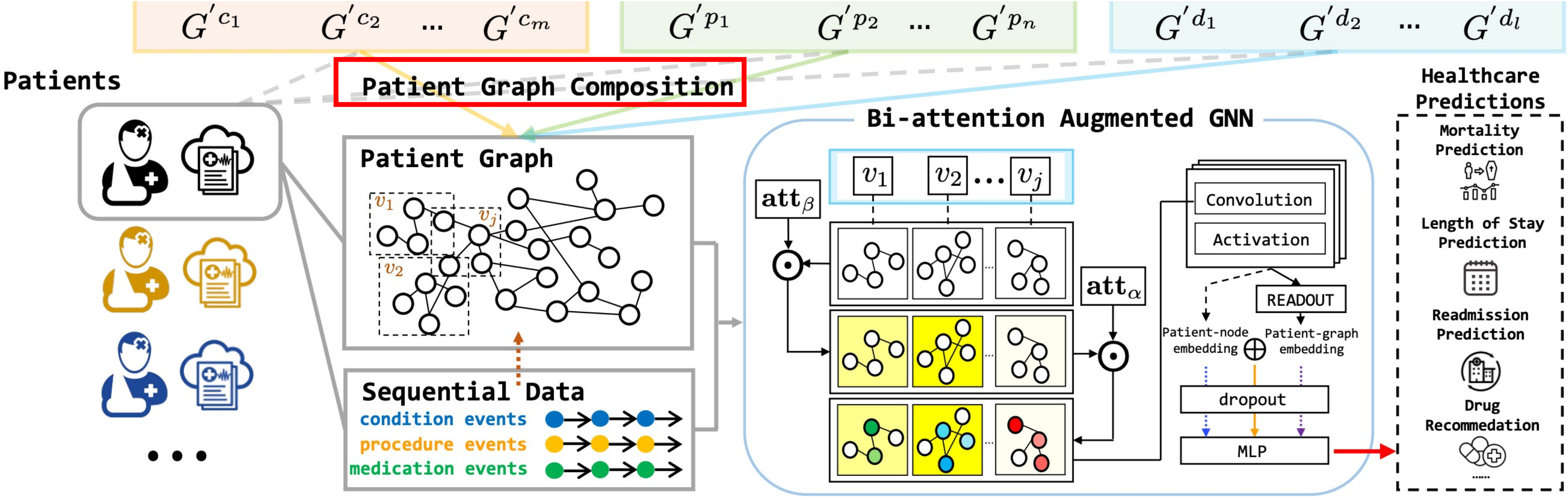
New concept-specific graph:

$$G'^{e} = (\mathcal{V}'^{e}, \mathcal{E}'^{e}) \subset G'$$

Method

GraphCare – Personalized KG Composition

Step 2: For each patient, merge relevant concept-specific KGs to form a personalized KG.

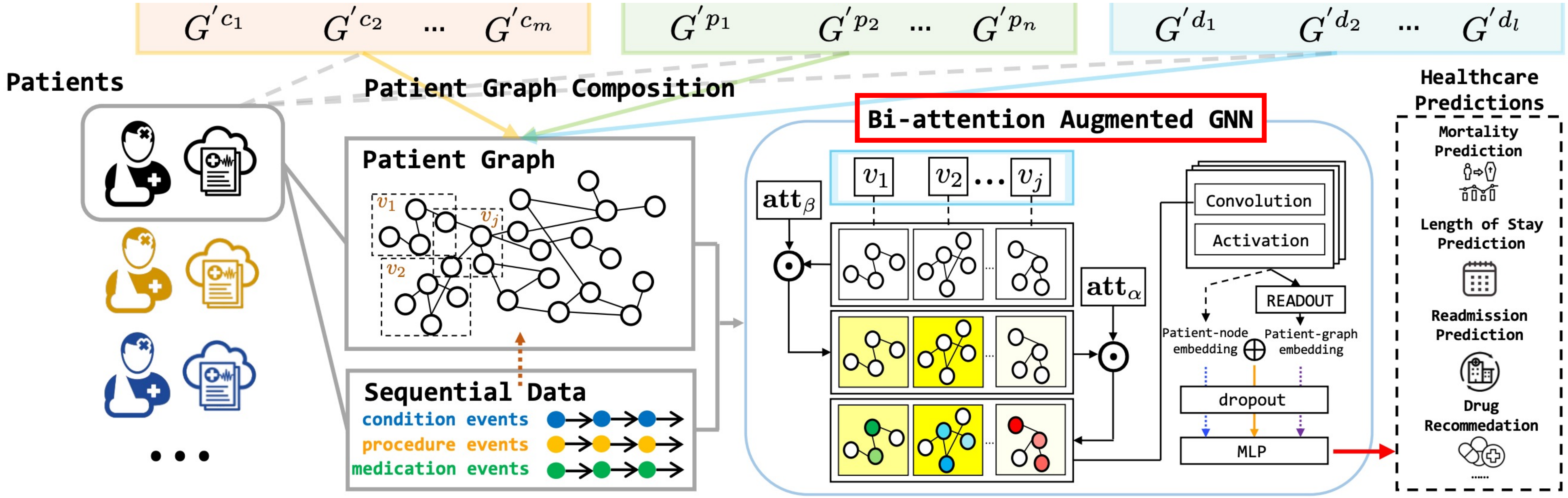


For each patient, we compose their personalized KG by merging the clustered KGs of their medical concepts. We create a patient node ($\mathcal{P}$) and connect it to its direct EHR nodes in the graph. The personalized KG for a patient can be represented as $G_{\text{pat}} = (\mathcal{V}_{\text{pat}}, \mathcal{E}_{\text{pat}})$, where $\mathcal{V}_{\text{pat}} = \mathcal{P} \cup \{\mathcal{V}'^{e_1}, \mathcal{V}'^{e_2}, ..., \mathcal{V}'^{e_\omega}\}$ and $\mathcal{E}_{\text{pat}} = \epsilon \cup \{\mathcal{E}'^{e_1}, \mathcal{E}'^{e_2}, ..., \mathcal{E}'^{e_\omega}\}$, with $\{e_1, e_2, ..., e_\omega\}$ being the medical concepts directly associated with the patient, ω being the number of concepts, and ϵ being the edge connecting $\mathcal{P}$ and $\{e_1, e_2, ..., e_\omega\}$. Further, as a patient is represented as a sequence of J visits (Choi et al., 2016a), the *visit-subgraphs* for patient i can be represented as $G_{\text{pat}(i)} = \{G_{i,1}, G_{i,2}, ..., G_{i,J}\} = \{(\mathcal{V}_{i,1}, \mathcal{E}_{i,1}), (\mathcal{V}_{i,2}, \mathcal{E}_{i,2}), ..., (\mathcal{V}_{i,J}, \mathcal{E}_{i,J})\}$ for visits $\{x_1, x_2, ..., x_J\}$ where $\mathcal{V}_{i,j} \subseteq \mathcal{V}_{\text{pat}(i)}$ and $\mathcal{E}_{i,j} \subseteq \mathcal{E}_{\text{pat}(i)}$ for $1 \leq j \leq J$.

Method

GraphCare – Bi-attention Augmented GNN

Step 3: Employ Bi-attention Augmented (BAT) Graph Neural Network (GNN) to predict based on the personalized KGs.



$$\alpha_{i,j,1},\ldots,\alpha_{i,j,M} = \mathrm{Softmax}(\mathbf{W}_\alpha \mathbf{g}_{i,j} + \mathbf{b}_\alpha),$$

$$\beta_{i,1},\ldots,\beta_{i,N} = \boldsymbol{\lambda}^\top \mathrm{Tanh}(\mathbf{w}_\beta^\top \mathbf{G}_i + \mathbf{b}_\beta), \quad \text{where} \quad \boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_N],$$

$$\mathbf{h}_{i,j,k}^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \sum_{j'\in J, k'\in\mathcal{N}(k)\cup\{k\}} \left(\underbrace{\alpha_{i,j',k'}^{(l)}\beta_{i,j'}^{(l)}\mathbf{h}_{i,j',k'}^{(l)}}_{\text{Node aggregation term}} + \underbrace{\mathrm{w}_{\mathcal{R}\langle k,k'\rangle}^{(l)}\mathbf{h}_{(i,j,k)\leftrightarrow(i,j',k')}}_{\text{Edge aggregation term}}\right) + \mathbf{b}^{(l)}\right)$$

Method

GraphCare – Bi-attention Augmented GNN

Step 3: Employ Bi-attention Augmented (BAT) Graph Neural Network (GNN) to predict based on the personalized KGs.

Attention Initialization

Table 4: Keyword candidates we attempted for attention initialization. We ***highlight*** the keywords we finally used in the experiments.

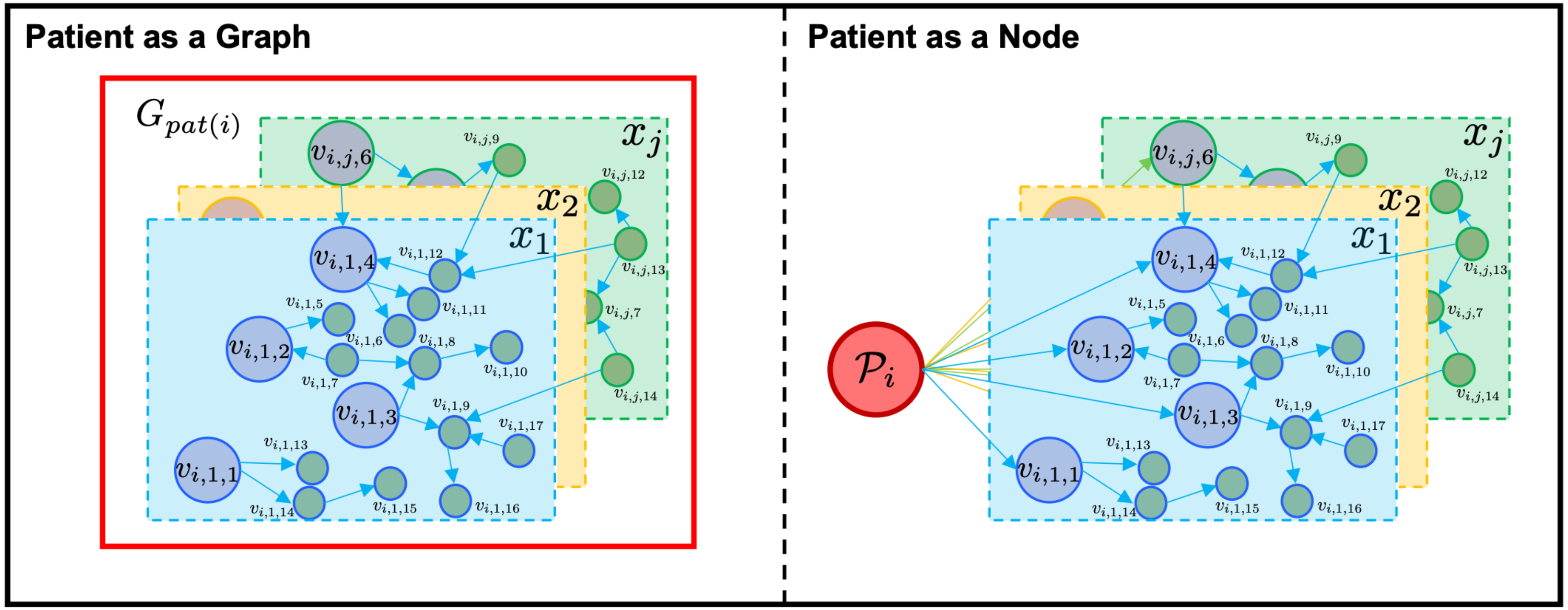
Task	Conditions	Procedures	Drugs
MT.	***terminal condition***, *critical diagnosis*, *end-stage*, *life-threatening*	***critical interventions***, *life-saving measures*, *resuscitation*, *emergency procedure*	*palliative medication*, *end-of-life drugs*, ***life support drugs***, *emergency meds*
RA.	*chronic ailment*, *postoperative complication*, *recurrent*, ***readmission-prone***	*follow-up procedure*, *secondary intervention*, ***post-treatment***, *treatment review*	*maintenance medication*, *postoperative drugs*, ***treatment continuation***, *follow-up meds*
LOS	*acute condition*, ***severe diagnosis***, *long-term ailment*, *extended-care diagnosis*	*major surgery*, ***intensive procedure***, *long recovery intervention*, *extended hospitalization*	-
Drug.	*chronic disease*, *acute diagnosis*, *symptomatic*, ***treatable condition***	*diagnostic procedure*, *treatment procedure*, ***medical intervention***, *drug-indicative procedure*	-

Method

GraphCare – Bi-attention Augmented GNN

Step 3: Employ Bi-attention Augmented (BAT) Graph Neural Network (GNN) to predict based on the personalized KGs.

Patient Representation



$$\mathbf{h}_i^{G_{\text{pat}}} = \text{MEAN}\left(\sum_{j=1}^{J}\sum_{k=1}^{K_j}\mathbf{h}_{i,j,k}^{(L)}\right), \quad \mathbf{h}_i^{\mathcal{P}} = \text{MEAN}\left(\sum_{j=1}^{J}\sum_{k=1}^{K_j}\mathbb{1}_{i,j,k}^{\Delta}\mathbf{h}_{i,j,k}^{(L)}\right),$$

$$\mathbf{z}_i^{\text{graph}} = \text{MLP}(\mathbf{h}_i^{G_{\text{pat}}}), \quad \mathbf{z}_i^{\text{node}} = \text{MLP}(\mathbf{h}_i^{\mathcal{P}}), \quad \mathbf{z}_i^{\text{joint}} = \text{MLP}(\mathbf{h}_i^{G_{\text{pat}}} \oplus \mathbf{h}_i^{\mathcal{P}})$$

Experiments

Performance comparison of 4 prediction tasks on MIMIC-III and MIMIC-IV

Findings:

1. GraphCare consistently outperforms other methods on all tasks and datasets.

2. BAT outperforms other GNNs and graph transformers.

3. Performance gain on MIMIC-III is more obvious.

Model		Task 1: Mortality Prediction MIMIC-III AUPRC	MIMIC-III AUROC	MIMIC-IV AUPRC	MIMIC-IV AUROC	Task 2: Readmission Prediction MIMIC-III AUPRC	MIMIC-III AUROC	MIMIC-IV AUPRC	MIMIC-IV AUROC
GRU		$11.8_{(0.5)}$	$61.3_{(0.9)}$	$4.2_{(0.1)}$	$69.0_{(0.8)}$	$68.2_{(0.4)}$	$65.4_{(0.8)}$	$66.1_{(0.1)}$	$66.2_{(0.1)}$
Transformer		$10.1_{(0.9)}$	$57.2_{(1.3)}$	$3.4_{(0.4)}$	$65.1_{(1.2)}$	$67.3_{(0.7)}$	$63.9_{(1.1)}$	$65.7_{(0.3)}$	$65.3_{(0.4)}$
RETAIN		$9.6_{(0.6)}$	$59.4_{(1.5)}$	$3.8_{(0.4)}$	$64.8_{(1.6)}$	$65.1_{(1.0)}$	$64.1_{(0.7)}$	$66.2_{(0.3)}$	$66.3_{(0.2)}$
GRAM		$11.4_{(0.7)}$	$60.4_{(0.9)}$	$4.4_{(0.3)}$	$66.7_{(0.7)}$	$67.2_{(0.8)}$	$64.3_{(0.4)}$	$66.1_{(0.2)}$	$66.3_{(0.3)}$
Deepr		$13.2_{(1.1)}$	$60.8_{(0.4)}$	$4.2_{(0.2)}$	$68.9_{(0.9)}$	$68.8_{(0.9)}$	$66.5_{(0.4)}$	$65.6_{(0.1)}$	$65.4_{(0.2)}$
AdaCare		$11.1_{(0.4)}$	$58.4_{(1.4)}$	$4.6_{(0.3)}$	$69.3_{(0.7)}$	$68.6_{(0.6)}$	$65.7_{(0.3)}$	$65.9_{(0.0)}$	$66.1_{(0.0)}$
GRASP		$9.9_{(1.1)}$	$59.2_{(1.4)}$	$4.7_{(0.1)}$	$68.4_{(1.0)}$	$69.2_{(0.4)}$	$66.3_{(0.6)}$	$66.3_{(0.3)}$	$66.1_{(0.2)}$
StageNet		$12.4_{(0.3)}$	$61.5_{(0.7)}$	$4.2_{(0.3)}$	$69.6_{(0.8)}$	$69.3_{(0.6)}$	$66.7_{(0.4)}$	$66.1_{(0.1)}$	$66.2_{(0.1)}$
GraphCare	w/ GAT	$14.3_{(0.8)}$	$67.8_{(1.1)}$	$5.1_{(0.1)}$	$71.8_{(1.0)}$	$71.5_{(0.7)}$	$68.1_{(0.6)}$	$67.4_{(0.4)}$	$67.3_{(0.4)}$
	w/ GINE	$14.4_{(0.4)}$	$67.3_{(1.3)}$	$5.7_{(0.1)}$	$72.0_{(1.1)}$	$71.3_{(0.8)}$	$68.0_{(0.4)}$	$68.3_{(0.3)}$	$67.5_{(0.4)}$
	w/ EGT	$15.5_{(0.5)}$	$69.1_{(1.0)}$	$6.2_{(0.2)}$	$71.3_{(0.7)}$	$72.2_{(0.5)}$	$68.8_{(0.4)}$	$68.9_{(0.2)}$	$67.6_{(0.3)}$
	w/ GPS	$15.3_{(0.9)}$	$68.8_{(0.8)}$	$\mathbf{6.7_{(0.2)}}$	$72.7_{(0.9)}$	$71.9_{(0.6)}$	$68.5_{(0.6)}$	$69.1_{(0.4)}$	$67.9_{(0.4)}$
	w/ BAT	$\mathbf{16.7_{(0.5)}}$	$\mathbf{70.3_{(0.5)}}$	$\mathbf{6.7_{(0.3)}}$	$\mathbf{73.1_{(0.5)}}$	$\mathbf{73.4_{(0.4)}}$	$\mathbf{69.7_{(0.5)}}$	$\mathbf{69.6_{(0.3)}}$	$\mathbf{68.5_{(0.4)}}$

Model		Task 3: Length of Stay Prediction MIMIC-III AUROC	MIMIC-III Kappa	MIMIC-III Accuracy	MIMIC-III F1-score	MIMIC-IV AUROC	MIMIC-IV Kappa	MIMIC-IV Accuracy	MIMIC-IV F1-score
GRU		$78.3_{(0.1)}$	$26.2_{(0.2)}$	$40.3_{(0.3)}$	$34.9_{(0.5)}$	$78.7_{(0.1)}$	$26.0_{(0.1)}$	$35.2_{(0.1)}$	$31.6_{(0.2)}$
Transformer		$78.3_{(0.2)}$	$25.4_{(0.4)}$	$40.1_{(0.3)}$	$34.8_{(0.2)}$	$78.3_{(0.3)}$	$25.3_{(0.4)}$	$34.4_{(0.2)}$	$31.4_{(0.3)}$
RETAIN		$78.2_{(0.1)}$	$26.1_{(0.4)}$	$40.6_{(0.3)}$	$34.9_{(0.4)}$	$78.9_{(0.3)}$	$26.3_{(0.2)}$	$35.7_{(0.2)}$	$32.0_{(0.2)}$
GRAM		$78.2_{(0.1)}$	$26.3_{(0.3)}$	$40.4_{(0.4)}$	$34.5_{(0.2)}$	$78.8_{(0.2)}$	$26.1_{(0.4)}$	$35.4_{(0.2)}$	$31.9_{(0.3)}$
Deepr		$77.9_{(0.1)}$	$25.3_{(0.4)}$	$40.1_{(0.6)}$	$35.0_{(0.4)}$	$79.5_{(0.3)}$	$26.4_{(0.2)}$	$35.8_{(0.3)}$	$32.3_{(0.1)}$
StageNet		$78.3_{(0.2)}$	$24.8_{(0.2)}$	$39.9_{(0.2)}$	$34.4_{(0.4)}$	$79.2_{(0.3)}$	$26.0_{(0.2)}$	$35.0_{(0.2)}$	$31.3_{(0.3)}$
GraphCare	w/ GAT	$79.4_{(0.3)}$	$28.2_{(0.2)}$	$41.9_{(0.2)}$	$36.1_{(0.4)}$	$80.3_{(0.3)}$	$28.4_{(0.4)}$	$36.2_{(0.1)}$	$33.3_{(0.3)}$
	w/ GINE	$79.2_{(0.2)}$	$28.3_{(0.3)}$	$41.5_{(0.3)}$	$36.0_{(0.4)}$	$79.9_{(0.2)}$	$27.5_{(0.3)}$	$36.3_{(0.3)}$	$32.8_{(0.2)}$
	w/ EGT	$80.3_{(0.3)}$	$28.8_{(0.2)}$	$42.8_{(0.4)}$	$36.3_{(0.5)}$	$80.5_{(0.2)}$	$28.7_{(0.3)}$	$36.7_{(0.2)}$	$33.5_{(0.1)}$
	w/ GPS	$80.9_{(0.3)}$	$28.8_{(0.4)}$	$43.0_{(0.3)}$	$36.8_{(0.4)}$	$80.7_{(0.3)}$	$28.8_{(0.4)}$	$36.7_{(0.3)}$	$33.9_{(0.3)}$
	w/ BAT	$\mathbf{81.4_{(0.3)}}$	$\mathbf{29.5_{(0.4)}}$	$\mathbf{43.2_{(0.4)}}$	$\mathbf{37.5_{(0.2)}}$	$\mathbf{81.7_{(0.2)}}$	$\mathbf{29.8_{(0.3)}}$	$\mathbf{37.3_{(0.3)}}$	$\mathbf{34.2_{(0.3)}}$

Model		Task 4: Drug Recommendation MIMIC-III AUPRC	MIMIC-III AUROC	MIMIC-III F1-score	MIMIC-III Jaccard	MIMIC-IV AUPRC	MIMIC-IV AUROC	MIMIC-IV F1-score	MIMIC-IV Jaccard
GRU		$77.0_{(0.1)}$	$94.4_{(0.0)}$	$62.3_{(0.3)}$	$47.8_{(0.3)}$	$74.1_{(0.1)}$	$94.2_{(0.1)}$	$60.2_{(0.2)}$	$44.0_{(0.4)}$
Transformer		$76.1_{(0.1)}$	$94.2_{(0.0)}$	$62.1_{(0.4)}$	$47.1_{(0.4)}$	$71.3_{(0.1)}$	$93.4_{(0.1)}$	$55.9_{(0.2)}$	$40.4_{(0.1)}$
RETAIN		$77.1_{(0.1)}$	$94.4_{(0.0)}$	$63.7_{(0.2)}$	$48.8_{(0.2)}$	$74.2_{(0.1)}$	$94.3_{(0.0)}$	$60.3_{(0.1)}$	$45.0_{(0.1)}$
GRAM		$76.7_{(0.1)}$	$94.2_{(0.1)}$	$62.9_{(0.3)}$	$47.9_{(0.3)}$	$74.3_{(0.2)}$	$94.2_{(0.1)}$	$60.1_{(0.2)}$	$45.3_{(0.3)}$
Deepr		$74.3_{(0.1)}$	$93.7_{(0.0)}$	$60.3_{(0.4)}$	$44.7_{(0.3)}$	$73.7_{(0.1)}$	$94.2_{(0.1)}$	$59.1_{(0.4)}$	$43.8_{(0.4)}$
StageNet		$74.4_{(0.1)}$	$93.0_{(0.1)}$	$61.4_{(0.3)}$	$45.8_{(0.4)}$	$74.4_{(0.1)}$	$94.2_{(0.0)}$	$60.2_{(0.3)}$	$45.4_{(0.4)}$
SafeDrug		$68.1_{(0.3)}$	$91.0_{(0.1)}$	$46.7_{(0.4)}$	$31.7_{(0.3)}$	$66.4_{(0.5)}$	$91.8_{(0.2)}$	$56.2_{(0.4)}$	$44.3_{(0.3)}$
MICRON		$77.4_{(0.0)}$	$94.6_{(0.1)}$	$63.2_{(0.4)}$	$48.3_{(0.4)}$	$74.4_{(0.1)}$	$94.3_{(0.1)}$	$59.3_{(0.3)}$	$44.1_{(0.3)}$
GAMENet		$76.4_{(0.0)}$	$94.2_{(0.1)}$	$62.1_{(0.4)}$	$47.2_{(0.4)}$	$74.2_{(0.1)}$	$94.2_{(0.1)}$	$60.4_{(0.4)}$	$45.3_{(0.3)}$
MoleRec		$69.8_{(0.1)}$	$92.0_{(0.1)}$	$58.1_{(0.1)}$	$43.1_{(0.3)}$	$68.6_{(0.1)}$	$92.1_{(0.1)}$	$56.3_{(0.4)}$	$40.6_{(0.3)}$
GraphCare	w/ GAT	$78.5_{(0.2)}$	$94.8_{(0.1)}$	$64.4_{(0.3)}$	$49.2_{(0.4)}$	$74.7_{(0.5)}$	$94.4_{(0.3)}$	$60.4_{(0.3)}$	$45.7_{(0.4)}$
	w/ GINE	$78.2_{(0.1)}$	$94.7_{(0.1)}$	$63.6_{(0.4)}$	$47.9_{(0.3)}$	$74.8_{(0.3)}$	$94.6_{(0.1)}$	$60.6_{(0.4)}$	$46.1_{(0.4)}$
	w/ EGT	$79.6_{(0.2)}$	$95.3_{(0.0)}$	$66.4_{(0.2)}$	$49.6_{(0.4)}$	$75.4_{(0.4)}$	$95.0_{(0.1)}$	$61.6_{(0.3)}$	$47.3_{(0.3)}$
	w/ GPS	$79.9_{(0.3)}$	$\mathbf{95.5_{(0.1)}}$	$66.2_{(0.3)}$	$\mathbf{49.8_{(0.4)}}$	$75.9_{(0.5)}$	$94.9_{(0.1)}$	$62.1_{(0.3)}$	$46.8_{(0.4)}$
	w/ BAT	$\mathbf{80.2_{(0.2)}}$	$\mathbf{95.5_{(0.1)}}$	$\mathbf{66.8_{(0.2)}}$	$49.7_{(0.3)}$	$\mathbf{77.1_{(0.1)}}$	$\mathbf{95.4_{(0.2)}}$	$\mathbf{63.9_{(0.3)}}$	$\mathbf{48.1_{(0.3)}}$

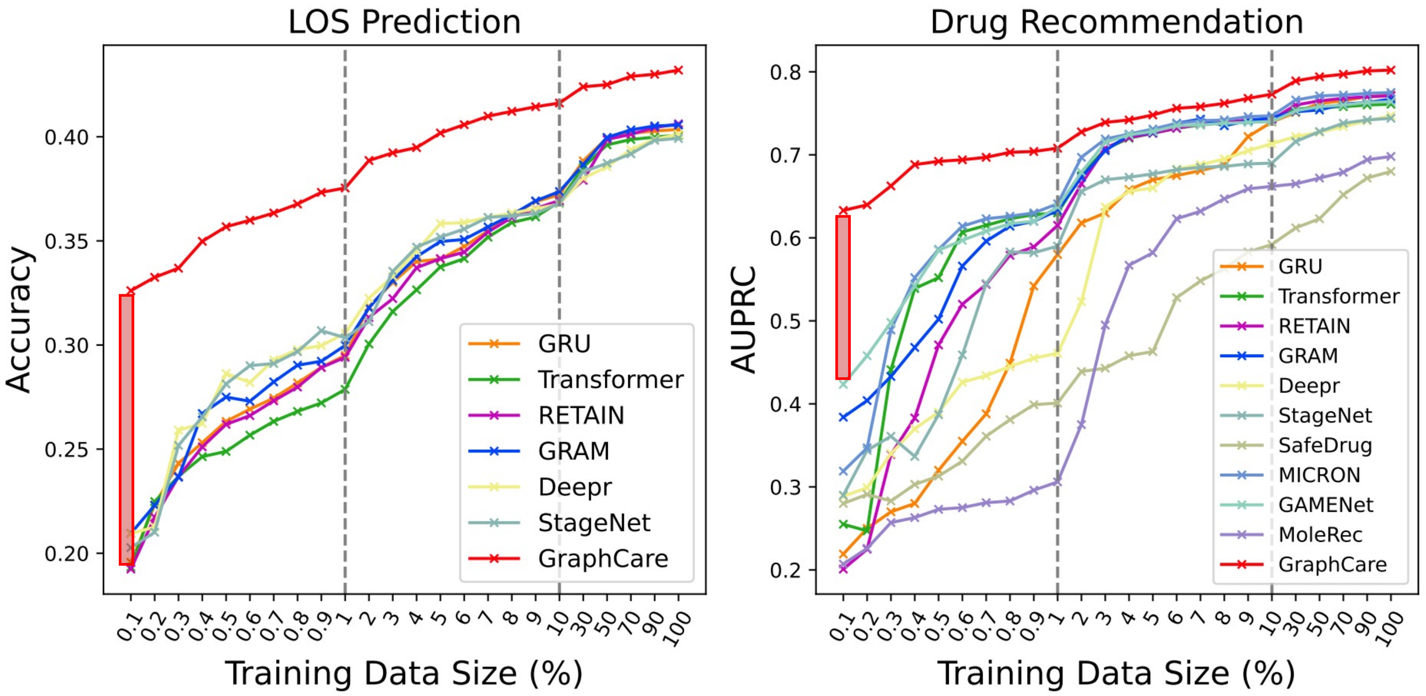
Experiments

Effect of EHR Training Data Size

Findings:

1. GraphCare exhibits a considerable edge over other models when confronted with scarce EHR training data.

2. Other graph-augmented methods (e.g. GRAM, GAMENet) also show a certain level of resilience against scarce data.



Experiments

Effect of Knowledge Graph Size

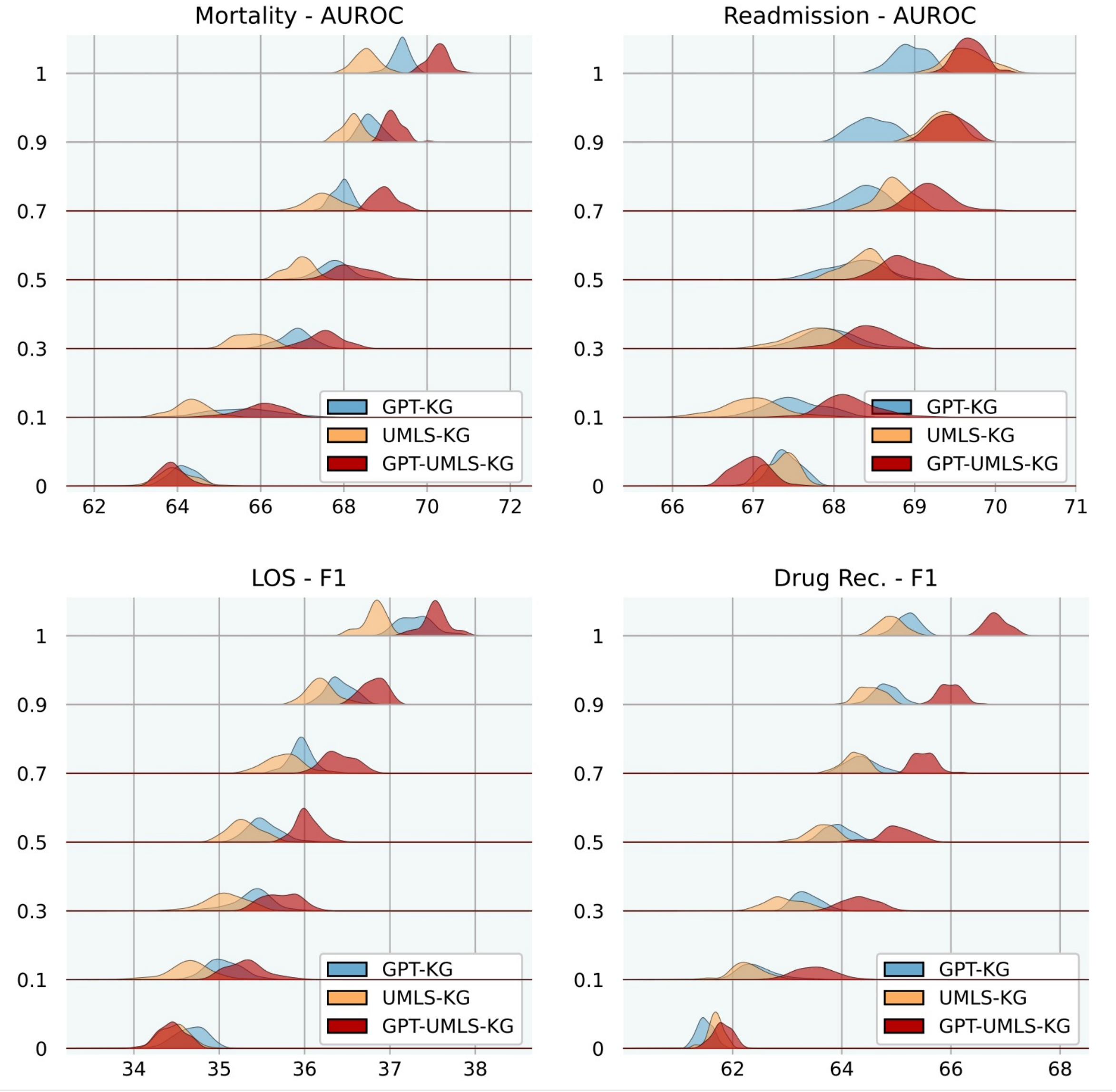
Statistics of Extracted KGs

KG	Hyperparameter	# Nodes	# Edges	# Triples
GPT-KG	χ=3	4599	752	31325
UMLS-KG	κ=1	3053	40	12421
UMLS-KG	κ=2, ϵ=5	10805	54	81073
GPT-UMLS-KG	χ=3, κ=1	6355	774	40496
GPT-UMLS-KG	χ=3, κ=2, ϵ=5	12284	785	104460

Findings:

1. Better performance with larger KGs.
2. Combined GPT-UMLS-KG outperforms both GPT-KG and UMLS-KG consistently.
3. GPT-KG contributes more to mortality and LOS predictions, while UMLS-KG edges out in readmission prediction.
4. Lower KG ratios are associated with larger standard deviations, due to the reduced likelihood of vital knowledge being contained.

Performance



(Based on 30 runs; Y-axis: ratio of KG used)

Experiments

Effect of BAT GNN

Statistics of MIMIC-III/IV

	#patients	#visits	#visits/patient	#conditions/patient	#procedures/patient	#drugs/patient
MIMIC-III	35,707	44,399	1.24	12.89	4.54	33.71
MIMIC-IV	123,488	232,263	1.88	21.74	4.70	43.89

Ablation Study of BAT GNN's Variants

		MIMIC-III				MIMIC-IV			
Case	Variants	**MT.**	**RA.**	**LOS**	**Drug.**	**MT.**	**RA.**	**LOS**	**Drug.**
#0	w/ *all*	70.3	69.7	37.5	66.8	73.1	68.5	34.2	63.9
#1	w/o α	$68.7_{\downarrow 0.6}$	$68.5_{\downarrow 1.2}$	$36.7_{\downarrow 0.8}$	$64.6_{\downarrow 2.2}$	$72.2_{\downarrow 0.9}$	$67.8_{\downarrow 0.7}$	$33.1_{\downarrow 1.1}$	$61.6_{\downarrow 2.3}$
#2	w/o β	$69.9_{\downarrow 0.4}$	$68.7_{\downarrow 1.0}$	$37.2_{\downarrow 0.3}$	$66.5_{\downarrow 0.3}$	$72.1_{\downarrow 1.0}$	$67.0_{\downarrow 1.5}$	$33.5_{\downarrow 0.7}$	$63.2_{\downarrow 0.7}$
#3	w/o $\mathrm{w}_{\mathcal{R}}$	$69.8_{\downarrow 0.5}$	$68.4_{\downarrow 1.3}$	$36.8_{\downarrow 0.7}$	$66.3_{\downarrow 0.5}$	$72.9_{\downarrow 0.2}$	$67.9_{\downarrow 0.6}$	$33.7_{\downarrow 0.5}$	$63.1_{\downarrow 0.8}$
#4	w/o *AttnInit*	$69.5_{\downarrow 0.8}$	$69.2_{\downarrow 0.5}$	$37.2_{\downarrow 0.3}$	$65.5_{\downarrow 1.3}$	$72.5_{\downarrow 0.6}$	$68.1_{\downarrow 0.4}$	$34.1_{\downarrow 0.1}$	$62.4_{\downarrow 1.5}$
#5	w/o #(1,2,3,4)	$67.4_{\downarrow 2.9}$	$68.1_{\downarrow 1.6}$	$36.0_{\downarrow 1.5}$	$64.0_{\downarrow 2.8}$	$71.7_{\downarrow 1.4}$	$67.5_{\downarrow 1.0}$	$32.9_{\downarrow 1.3}$	$60.5_{\downarrow 3.4}$

Observations:

1. Excluding node-level attention (α) results in a general drop across all tasks/datasets.
2. Excluding visit-level attention (β) affects more on MIMIC-IV, as it has a higher #visits/patient.
3. Readmission prediction is more sensitive to the visit-level attention.
4. Drug recommendation is more sensitive to the attention initialization.

Experiments

Appendix: Patient Representation Learning

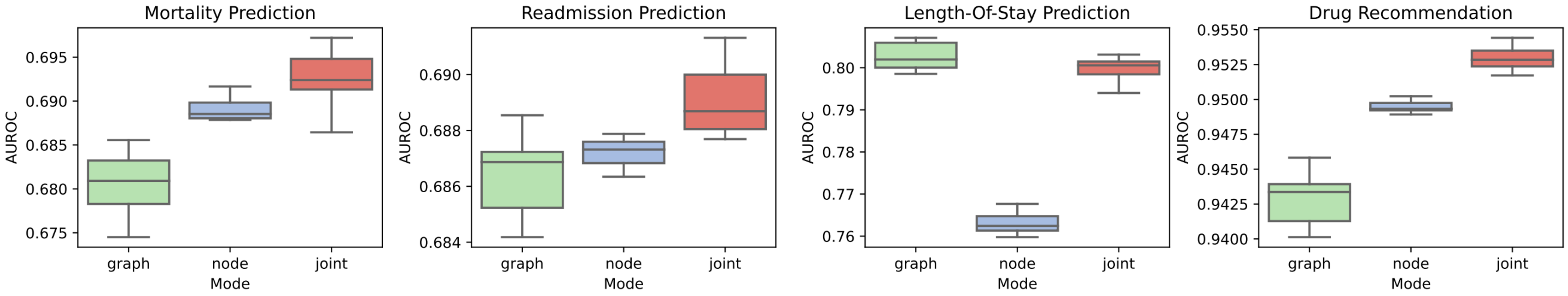


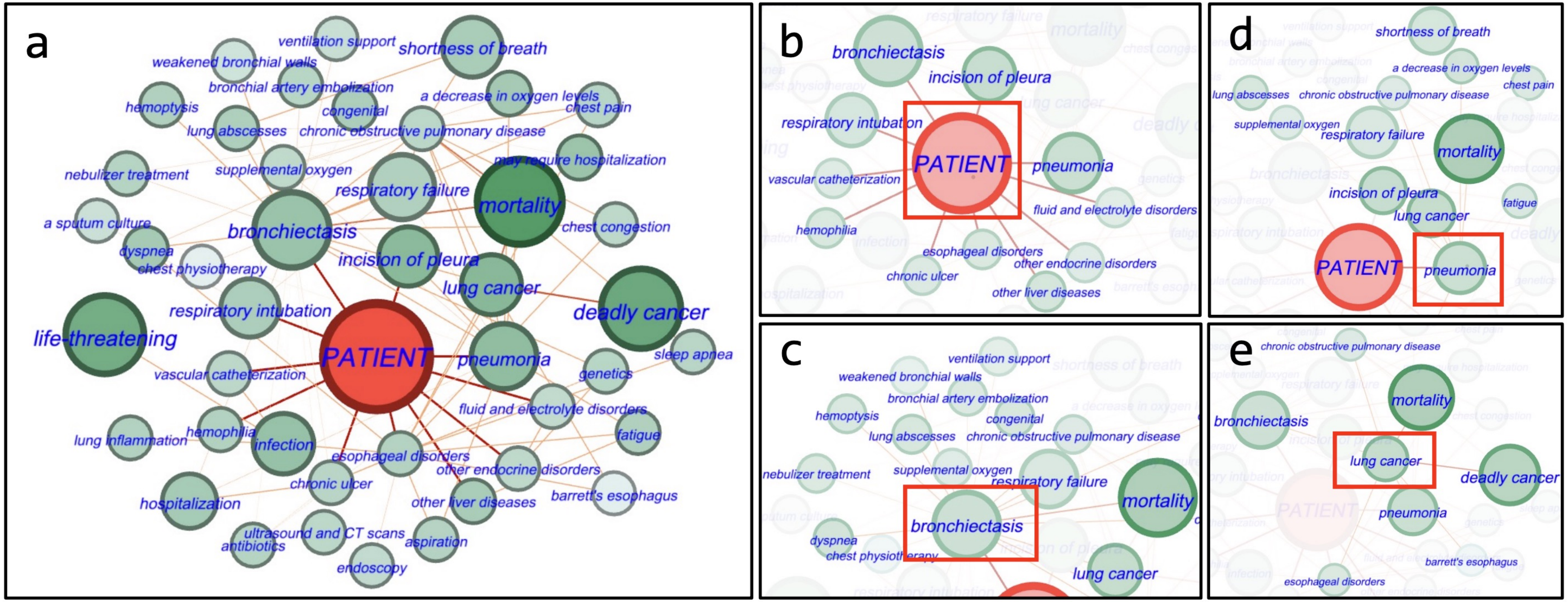
Figure 8: **Performance of healthcare predictions with three types of patient representations (§3.3)**: (1) **graph** - patient graph embedding obtained through mean pooling of node embedding; (2) **node** - patient node embedding connected to the direct EHR node; (3) **joint** - embedding concatenated by (1) and (2). We use GPT-KG to perform this analysis.

Observations:

The patient graph contains more information as well as more noise.
The patient node contains more accurate information, as it directly links the EHR nodes.
A joint representation of them is a balance.

Experiments

Interpretability of GraphCare



Observations:

Removing the indirect node “lung cancer” connecting crucial nodes “mortality” and “deadly cancer” would result in a failure of mortality prediction.

➡ Highlight the importance of the personalized KG.

Thank you!

Preprint: https://arxiv.org/pdf/2305.12788.pdf
Code: https://github.com/pat-jj/GraphCare