REFINE: LLM Refinement over Budgeted Text-Attributed Graphs for Personalized Medical Concept Representation

Mohsen Nayebi Kerdabadi Arya Hadizadeh Moghaddam Dongjie Wang Zijun Yao
University of Kansas, Lawrence, KS

EMNLP 2026

Read the paper

Why personalize medical concepts?

The same medical code can mean stable disease in one patient and acute deterioration in another.

Relational challenge. Which KG context does each code need? Uniform expansion can add irrelevant neighbors.

Semantic challenge. How can patient history and selected KG evidence refine a code's meaning?

REFINE learns a representation for each patient–code context.

What REFINE contributes

Sequential RL budgeting adaptively chooses expansion depth and fanout for each observed code, accounting for context already selected.

Graph-aware LLM refinement combines trainable soft tokens with a frozen LLM and a parallel heterogeneous GNN.

The fused representations plug into standard EHR backbones.

Methodology

1. Global TKG and patient-personalized graph construction

Global TKG. Filter code pairs by co-occurrence and next-visit statistics. Type-constrained LLM prompts supply relations and rationales; descriptions provide node semantics.

Patient graph. Link patients, visits, and codes; add selected KG neighbors and typed edges using only pre-cutoff history.

2. Personalized concept sub-KG extraction via sequential RL

Action. Choose a code root c and an expansion $a = (K, \mathbf{m})$: hop depth and per-hop fanout. Options range from no expansion to two hops. Evidence scores rank neighbors within the chosen budget.

State. Condition each decision on patient history, KG context already selected, and the remaining budget. A macro-action updates one code's expansion.

Sequential selection. Later choices account for context added for earlier codes. Stop when the policy chooses STOP, exhausts the budget, or reaches the step limit.

$$R_p = \mathcal{L}_p(G_p^0) - \mathcal{L}_p(G_p^{\text{sel}}) - \lambda_{\text{cost}} C_p$$

Reward: prediction-loss improvement over no expansion, minus a cost for selected KG nodes and edges.

Training. Warm up and freeze a GNN-only proxy, then train an actor–critic selector. Use greedy selection at inference and cache the selected graphs for LLM refinement.

3. Graph-aware LLM refinement for personalized concept representation

Two complementary soft tokens. A trainable projection maps the code embedding, blended with fixed description semantics, to a code token $s_{p,c}$. A lightweight GNN-style aggregator pools RL-selected neighbors and relation/rationale embeddings into a KG-context token $k_{p,c}$.

Graph-aware mixed prompt. Interleave both soft tokens with hard text for the task, visits, code names, and types (Figure 2). Each visit–code block is

$$\Gamma(p, t, c) = [\text{name}(c), \phi(c), \langle s_{p,c} \rangle, \langle k_{p,c} \rangle].$$

Frozen-LLM refinement. Encode the mixed sequence X_p and project the hidden state at the code-soft-token position $i(p, t, c)$:

$$H_p^L = \text{LLM}(X_p), \qquad h_{p,t,c}^L = W_L H_{p,i(p,t,c)}^L.$$

Shared input soft tokens yield **visit-specific outputs**: each occurrence attends to its patient history and selected KG evidence. Soft-token components are trainable; the LLM stays frozen.

REFINE at a glance

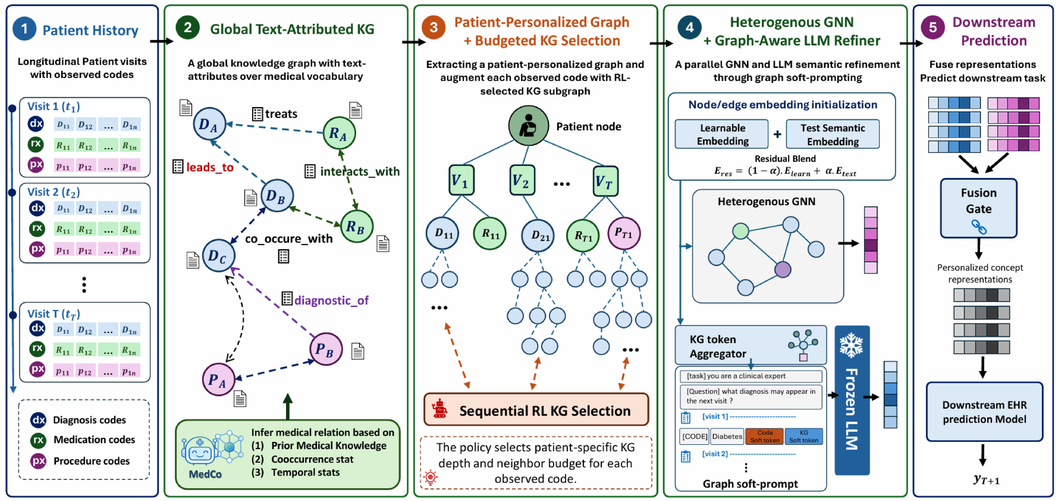

Figure 1. REFINE framework. Sequential RL selects patient-specific KG context. Parallel GNN and frozen-LLM branches refine and fuse concept representations for prediction.

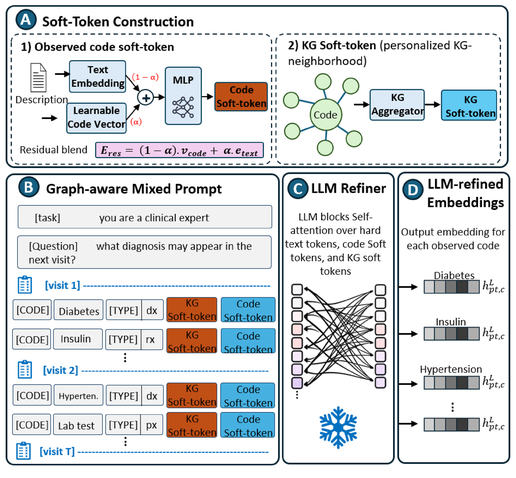

Figure 2. Graph-aware soft prompting. Code and KG-context soft tokens guide frozen-LLM refinement within a visit-structured prompt.

4. Fusion of GNN and LLM representations and integration with downstream models

Complementary context. The heterogeneous GNN encodes patient-specific graph structure as $h_{p,c}^G$. The frozen LLM produces visit-specific semantic context $h_{p,t,c}^L$ from the mixed prompt.

Learned fusion gate. A trainable gate balances the structural and semantic representations for each visit–code occurrence:

$$g_{p,t,c} = \sigma\big(W_F[h_{p,c}^G; h_{p,t,c}^L]\big),$$
$$z_{p,t,c} = g_{p,t,c} \odot h_{p,t,c}^L + (1 - g_{p,t,c}) \odot h_{p,c}^G.$$

Plug-in concept encoder. Pool the fused code representations within each visit, then pass the visit sequence to a standard EHR backbone for next-visit diagnosis prediction.

$$h_p = f_\theta\big(\{\text{Pool}(\{z_{p,t,c} : c \in V_{p,t}\})\}_{t=1}^{t^*}\big), \qquad \hat{y}_p = \sigma(W_o h_p + b_o).$$

Stage II training. With RL-selected graphs fixed, optimize the graph components, soft-token projections, fusion gate, and downstream predictor using multi-label binary cross-entropy. The LLM backbone remains frozen.

Backbone compatibility. REFINE replaces code embeddings while retaining each EHR backbone's visit-sequence modeling.

Evaluation setup

We evaluate next-visit diagnosis prediction on **MIMIC-III** (7,515 patients; 12,430 visit samples) and **MIMIC-IV** (18,829 patients; 25,028 visit samples). We report five-fold average AUPRC, F1, and Acc@k, using a frozen Llama-3.2-1B refiner.

Consistent gains across five EHR backbones

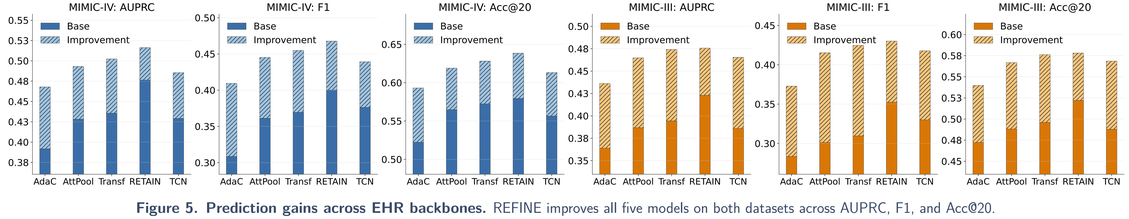

Figure 5. Prediction gains across EHR backbones. REFINE improves all five models on both datasets across AUPRC, F1, and Acc@20.

Main results: next-visit prediction

Table 1. Next-visit diagnosis prediction. Selected encoders with the same Transformer backbone. Overall metrics precede label-frequency quartiles; all values are percentages.

	Overall			Label-Frequency AUPRC			
Encoder	AUPRC	F1	Acc@20	0–25%	25–50%	50–75%	75–100%
MIMIC-III							
Base	39.44	30.96	49.60	38.64	47.30	70.63	74.30
GRAM	40.10	32.32	51.02	40.06	49.84	71.78	75.16
GraphCare	43.35	35.46	56.00	44.80	58.16	70.97	65.72
Rel-LLM	43.58	36.05	56.08	44.95	58.35	73.85	78.40
LINKO	44.10	37.05	56.20	45.20	59.05	74.25	79.20
MedCO	45.35	39.90	56.32	45.73	64.00	73.80	82.80
REFINE	**47.41**	**42.46**	**57.62**	**47.92**	**66.55**	**76.28**	**86.95**
MIMIC-IV							
Base	43.56	36.94	57.24	44.93	53.50	51.85	73.10
GRAM	44.72	37.78	58.02	45.76	54.39	52.60	73.80
GraphCare	46.90	40.12	59.14	48.31	**60.31**	**73.70**	66.33
Rel-LLM	47.20	40.55	59.85	48.70	56.80	56.40	76.60
LINKO	47.55	41.10	60.55	49.10	57.10	56.95	77.20
MedCO	48.28	43.10	61.55	49.95	57.35	57.70	78.20
REFINE	**50.24**	**45.48**	**62.82**	**52.30**	57.98	59.65	**81.55**

AUPRC gain over Base: +7.97 points on MIMIC-III and +6.68 points on MIMIC-IV.

Which components matter?

Table 2. REFINE component ablations on MIMIC-IV. All rows and metrics from the paper's Table 3 (%).

Variant	AUPRC	F1	Acc@20
Full REFINE	**50.24**	**45.48**	**62.82**
w/o RL budgeting	49.18	44.18	61.92
w/o full LLM refiner	49.32	44.05	61.84
w/o soft prompt tokens	49.38	44.12	61.90
w/o KG soft tokens	49.72	44.78	62.32
w/o full personalized GNN	47.35	40.82	60.25
Global code representation (MedCo)	48.28	43.10	61.55

The personalized GNN has the largest ablation effect; RL selection and LLM refinement both improve AUPRC.

KG selection and policy behavior

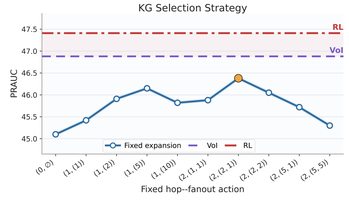

Figure 3. KG selection strategies on MIMIC-III. Sequential code-level RL outperforms fixed budgets and visit-level value-of-information selection.

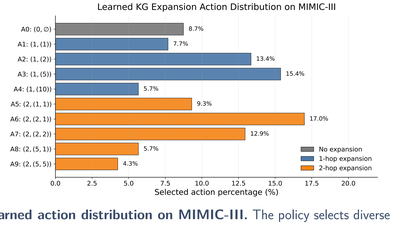

Figure 4. Learned action distribution on MIMIC-III. The policy selects diverse no-expansion, one-hop, and two-hop budgets.

Takeaway

REFINE combines sequential RL selection with graph-aware LLM soft prompting to learn patient-specific concept representations. The policy adapts KG depth and neighbor budgets for each observed code, while the frozen LLM refines code embeddings using personalized graph context. Together, these components improve next-visit diagnosis prediction across two datasets and five EHR backbones.