GraphWalker: Patient Analogy Meets Information Gain for Clinical Reasoning with Large Language Models

Anonymous ACL submission

Abstract

Clinical reasoning over electronic health records (EHRs) is a fundamental yet challenging task in modern healthcare. While large language models (LLMs) offer a promising paradigm via in-context demonstrations that requires no task-specific parameter updates, existing methods for **reasoning by patient analogy** in EHR settings suffer from three core limitations: **(1) Perspective Limitation**, where data-driven similarity misaligns with LLM reasoning needs while model-driven signals are constrained by limited clinical competence; **(2) Cohort Awareness**, as demonstrations are selected independently without modeling population-level structure; and **(3) Information Aggregation**, where redundancy and interaction effects among demonstrations are ignored. We propose *GraphWalker*, a training-free framework that lets frozen LLMs **reason by analogy over retrieved patient cases**. *GraphWalker* (i) jointly leverages data-driven and model-driven perspectives, (ii) discovers patient cohorts to ground retrieval in population-level structure, and (iii) employs a lazy greedy search with frontier expansion to compose demonstrations with high marginal information gain. Extensive experiments on multiple real-world EHR benchmarks show that *GraphWalker* consistently outperforms state-of-the-art demonstration selection baselines, and remains substantially more robust under cross-dataset distribution shift, without task-specific parameter updates. *GraphWalker* further generalizes to black-box LLMs and composes naturally with agentic reasoning frameworks, positioning it as a pluggable **patient-analogy skill** in LLM-based clinical workflows. Our code is available at `https://anonymous.4open.science/r/GraphWalker-4473`.

1 Introduction

Clinical reasoning on electronic health records (EHRs) underpins critical applications such as diagnosis support, risk stratification, and treatment planning (Ma et al., 2023; Xu et al., 2023a,b, 2024b; Fang et al., 2023; Liao et al., 2025). Recent large language models (LLMs) have been adapted for EHR clinical reasoning via pre-training (Zhang et al., 2025a), post-training (Ding et al., 2025a; Fang et al., 2025; Ding et al., 2025b), or hybrid pipelines that distill LLM-derived knowledge into supervised EHR models (Jiang et al., 2024; Xu et al., 2025). However, these approaches rely on parameter updates over task-specific labeled data, with their effectiveness constrained by fixed input schemas that are brittle to variations in feature order, availability, and encoding across heterogeneous EHR systems (Brown et al., 2024; Chen et al., 2024; Fang et al., 2025). Yet clinical practice itself relies on a fundamentally different mode of reasoning: physicians make decisions by recalling and comparing similar past patients, an approach known as *reasoning by analogy* (Norman, 2006; Ten Cate et al., 2017). This motivates a lightweight inference-time strategy: equipping a frozen LLM to retrieve and reason over similar past patients, avoiding task-specific parameter updates while remaining robust to distribution shift and composable as a patient-analogy skill in LLM-based clinical workflows.

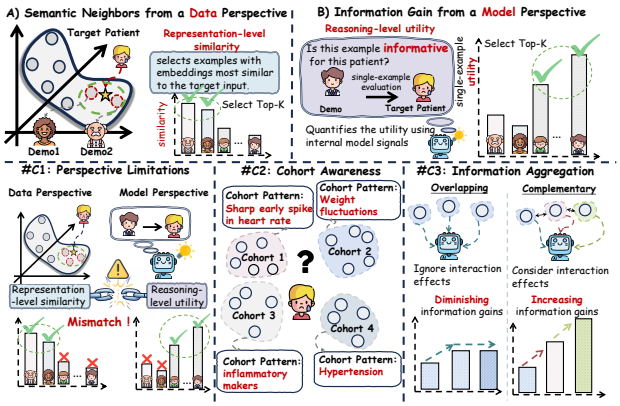

Figure 1: Illustration of existing demonstration selection methods (top) and their three core challenges (bottom).

Existing demonstration selection methods fall

into two perspectives.

- ***Semantic Neighbors*** *from a Data Perspective*: As illustrated in Fig 1(A), this line (Liu et al., 2022; Hongjin et al., 2022; Robertson et al., 2009) leverages a pre-trained representation model (e.g., SMART) (Yu et al., 2024) to embed patient records into a latent space, and retrieves the most similar past patients as in-context demonstrations.
- ***Information Gain*** *from a Model Perspective*: As illustrated in Fig 1(B), this line (Gonen et al., 2023; Peng et al., 2024; Liu et al.; Li et al., 2025) quantifies the utility of candidate demonstrations using LLM-internal signals, such as conditional entropy reduction or gradient-based influence, and selects the top-k demonstrations that maximally improve the model's predictive confidence.

Through a systematic analysis of these methods (Section 2), we identify **three core challenges** shared by existing works, as shown in Fig 1(C).

- *Perspective Limitation*: ***Semantic Neighbors*** implicitly assume that embedding similarity aligns with the LLM's reasoning needs, an assumption that does not consistently hold: highly similar patient trajectories may still contribute little to downstream predictions, suggesting a mismatch between representation-level similarity and reasoning-level utility. ***Information Gain***, although more directly aligned with the target model, is inherently constrained by the LLM's clinical competence; when the model lacks sufficient domain knowledge, its internal uncertainty signals may fail to reflect clinically meaningful longitudinal patterns.
- *Cohort Awareness*: ***Both perspectives*** focus on retrieving isolated similar patients, treating each demonstration as independent. However, clinical reasoning is inherently population-driven; selecting individual neighbors without modeling cohort structure can amplify noise from idiosyncratic cases and lead to unstable reasoning contexts.
- *Information Aggregation*: ***Both perspectives*** typically assume that the utility of selected demonstrations accumulates linearly. In practice, multiple similar demonstrations often encode overlapping clinical signals, yielding diminishing marginal gains; ignoring redundancy and interaction effects wastes the limited context window.

These challenges motivate a principled framework that bridges data and model perspectives, captures cohort structure, and accounts for demonstration interactions. To this end, we propose *GraphWalker*, a graph-guided framework for **Reasoning by Patient Analogy**, enabling LLMs to retrieve and reason over clinically similar patients without task-specific parameter updates. *GraphWalker integrates data and model perspectives* by encoding patient records through a pre-trained EHR representation model to construct a patient cohort graph, while leveraging LLM-estimated information gain to guide traversal and demonstration selection. To capture cohort awareness, *GraphWalker* replaces instance-level retrieval with *Cohort Discovery*, enabling reasoning over clinically coherent patient groups rather than isolated cases. To address redundancy and interaction among retrieved demonstrations, we introduce *Lazy Greedy Search with Frontier Expansion*, formulating demonstration selection as a combinatorial optimization problem that approximates locally optimal demonstration sets under a constrained context budget.

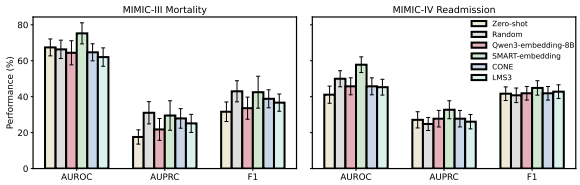

Figure 2: Analysis of the Limitations of a Single Perspective.

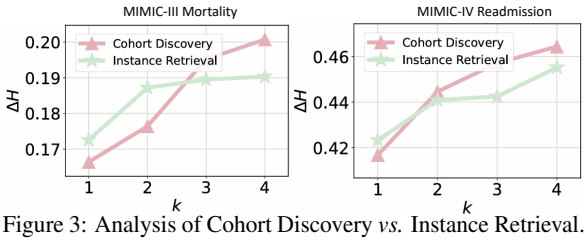

Figure 3: Analysis of Cohort Discovery *vs.* Instance Retrieval.

To summarize, our contributions are threefold:

- **Insightfully**, through a systematic analysis of existing LLM-based clinical reasoning approaches, we identify three core challenges in demonstration selection for analogical patient reasoning: *Perspective Limitation*, *Cohort Awareness*, and *Information Aggregation*.
- **Technically**, we propose *GraphWalker*, a graph-guided framework for reasoning by patient analogy that (1) jointly leverages clinical similarity and LLM-estimated information gain by *integrating data and model perspectives*; (2) employs *Cohort Discovery* to ground retrieval in clinically coherent patient groups rather than isolated cases; and (3) introduces *Lazy Greedy Search with Frontier Expansion* to mitigate diminishing returns in demonstration aggregation.

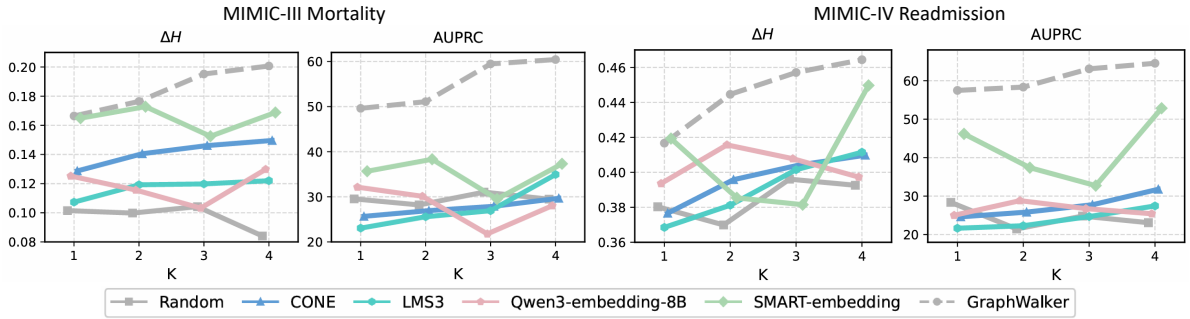

Figure 4: Analysis of information aggregation across multiple demonstrations under different demonstration selection algorithms.

- **Experimentally**, on multiple real-world EHR benchmarks, *GraphWalker* consistently outperforms state-of-the-art demonstration selection baselines, outperforms strong supervised EHR predictors in-domain, and remains substantially more robust under cross-dataset distribution shift, all with only seconds-level inference overhead. *GraphWalker* further generalizes to black-box LLMs and composes naturally with agentic reasoning frameworks, positioning it as a pluggable patient-analogy skill in clinical workflows.

2 Observation & Motivation

In this section, we conduct a systematic analysis of existing demonstration selection methods and present key observations that motivate the design of *GraphWalker*. Detailed experimental settings are provided in Appendix E.

2.1 Motivation on Perspective Limitation

We compare representative methods from both perspectives: **Semantic Neighbors** from a data perspective, instantiated as Qwen3-Embedding-8B (Yang et al., 2025) and SMART (Yu et al., 2024) (pre-trained on a large-scale EHR corpus); and **Information Gain** from a model perspective, instantiated as CONE (Peng et al., 2024) and LMS3 (Liu et al.). All methods are evaluated against Zero-shot and Random baselines using the same prompt.

As shown in Figure 2, SMART achieves the best performance in most cases, confirming the importance of clinical knowledge in demonstration selection. Meanwhile, CONE and LMS3 consistently outperform the generic Qwen3-Embedding-8B, suggesting that information gain serves as an effective selection criterion even in the absence of explicit clinical knowledge. However, neither perspective dominates: each yields strong results on some metrics but unstable performance on others. This complementary yet inconsistent behavior indicates that no single perspective fully captures the signals needed for effective demonstration selection, motivating an integrated approach.

> **Observation I.** *Both clinical knowledge (Data Perspective) and information gain (Model Perspective) provide valuable but incomplete guidance for demonstration selection; effective selection requires integrating both perspectives.*

2.2 Motivation on Cohort Awareness

We compare traditional instance retrieval with our proposed *Cohort Discovery* strategy. Using SMART as the EHR encoder, we construct a patient graph: instance retrieval selects the most similar patient and performs local walks to identify remaining demonstrations, whereas *Cohort Discovery* first locates the most relevant cohort and then selects demonstrations from within it.

As shown in Figure 3, instance retrieval is effective when k is small but becomes less competitive than *Cohort Discovery* as k increases. We attribute this to the following: a single most-similar patient may be an outlier or lie in a sparse cluster, biasing subsequent selection toward local optima. In contrast, *Cohort Discovery* leverages the relational structure among patients, providing a more robust basis for multi-demonstration selection.

> **Observation II.** *Instance retrieval overemphasizes a single nearest demonstration during initial selection and is prone to local optima; as the number of demonstrations increases, its performance degrades relative to cohort-based approaches.*

2.3 Motivation on Information Aggregation

We further analyze how existing methods behave in the multi-demonstration setting. As an analysis metric, we introduce the LLM's conditional entropy reduction (ΔH, see Appendix E), which captures how much a given demonstration composition reduces the model's predictive uncertainty on the target patient.

Figure 4 reveals two key findings. **(1)** Task performance closely tracks the trend of ΔH, indicating that ΔH serves as an effective guiding signal for demonstration selection. This further strengthens our motivation to integrate both data-driven and model-driven perspectives. **(2) Semantic Neighbor** methods overlook information aggregation: in multi-demonstration settings, their performance degrades due to information redundancy and conflicts among selected demonstrations. **Information Gain** methods, in contrast, maintain monotonic performance improvement as more demonstrations are added, yet exhibit pronounced diminishing marginal returns due to insufficient consideration of the demonstration set as a whole—each demonstration is scored independently without modeling interactions with previously selected ones.

> **Observation III.** *Both **Semantic Neighbor** and **Information Gain** approaches insufficiently account for Information Aggregation, leading to performance degradation or diminishing marginal returns in multi-demonstration settings.*

Summary. Based on the above observations, we propose *GraphWalker*, which addresses three key challenges in demonstration selection through targeted strategies. Specifically, we (1) *integrate data and model perspectives* to aggregate clinical knowledge and leverage information gain as a guiding signal (for ***Observation I***), (2) replace *Instance Retrieval with Cohort Discovery*, enabling reasoning over clinically coherent patient groups rather than isolated cases (for ***Observation II***), and (3) introduce a *Lazy Greedy Search with Frontier Expansion*, which formulates demonstration selection as a combinatorial optimization problem to address Information Aggregation (for ***Observation III***).

3 Methodology

As illustrated in Figure 5, *GraphWalker* operates in three stages:

- **Patient Graph Construction and Cohort Discovery** builds a population-level patient graph and partitions it into coherent cohorts, capturing shared clinical patterns beyond isolated patients.
- **Instance-aware Cohort Retrieval and Anchor Initialization** retrieves clinically relevant cohorts for the target patient and initializes anchor nodes within them, providing a focused candidate set.
- **LLM-guided Lazy Greedy Search with Frontier Expansion** iteratively constructs the case set by selecting frontier nodes that maximize marginal information gain conditioned on the current composition.

3.1 Patient Graph Construction and Cohort Discovery

Patient Representation. To capture clinical patterns that generic semantic embeddings overlook (see Section 2), we adopt a pretrained Transformer-based EHR encoder (e.g., *SMART* (Yu et al., 2024)). Formally, each patient's EHR record is a sequence of T time-ordered visits $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$. Given an EHR base of N patient records $\{\mathbf{X}_i\}_{i=1}^{N}$ (i.e., the training set), the encoder $\mathcal{M}_{\text{exp}}$ maps each record to a fixed-dimensional embedding $\mathbf{h}_i = \mathcal{M}_{\text{exp}}(\mathbf{X}_i) \in \mathbb{R}^d$, which serves as the patient representation for subsequent graph construction.

Population-level Graph Construction. We construct a k-nearest-neighbor graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over patient embeddings to encode local clinical similarity at the population level. Each node $v_i \in \mathcal{V}$ corresponds to a patient, and an edge $(i, j) \in \mathcal{E}$ is established iff patient j is among the k_g nearest neighbors of patient i under cosine similarity, i.e., $\mathcal{E} = \{(i, j) \mid \text{rank}_j \, \text{sim}(\mathbf{h}_i, \mathbf{h}_j) \leq k_g\}$, where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and k_g controls graph density.

Graph-based Cohort Discovery. To capture shared clinical patterns at the cohort level, we partition the patient graph into coherent cohorts of similar patients. Specifically, we adopt the *Leiden* algorithm (Traag et al., 2019) because its well-connected community partitions are well suited for constructing stable cohort prototypes and supporting subsequent graph-based frontier expansion. This produces a set of cohort subgraphs $\{\mathcal{C}_m\}_{m=1}^{M}$ by maximizing graph modularity $\mathcal{Q}$:

$$\mathcal{Q} = \frac{1}{2|\mathcal{E}|} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2|\mathcal{E}|} \right) \mathbb{I}(c_i = c_j), \qquad (1)$$

where A_{ij} is the adjacency matrix of $\mathcal{G}$, d_i is the degree of node i, and $\mathbb{I}(\cdot)$ indicates whether two nodes belong to the same cohort. Each cohort exhibits dense internal connectivity and sparse external connectivity. We summarize each cohort $\mathcal{C}_m$ by mean-pooling its member embeddings into a cohort prototype $\mathbf{z}_m = \frac{1}{|\mathcal{C}_m|} \sum_{i \in \mathcal{C}_m} \mathbf{h}_i$, which serves as a compact cohort representation for instance-aware cohort retrieval.

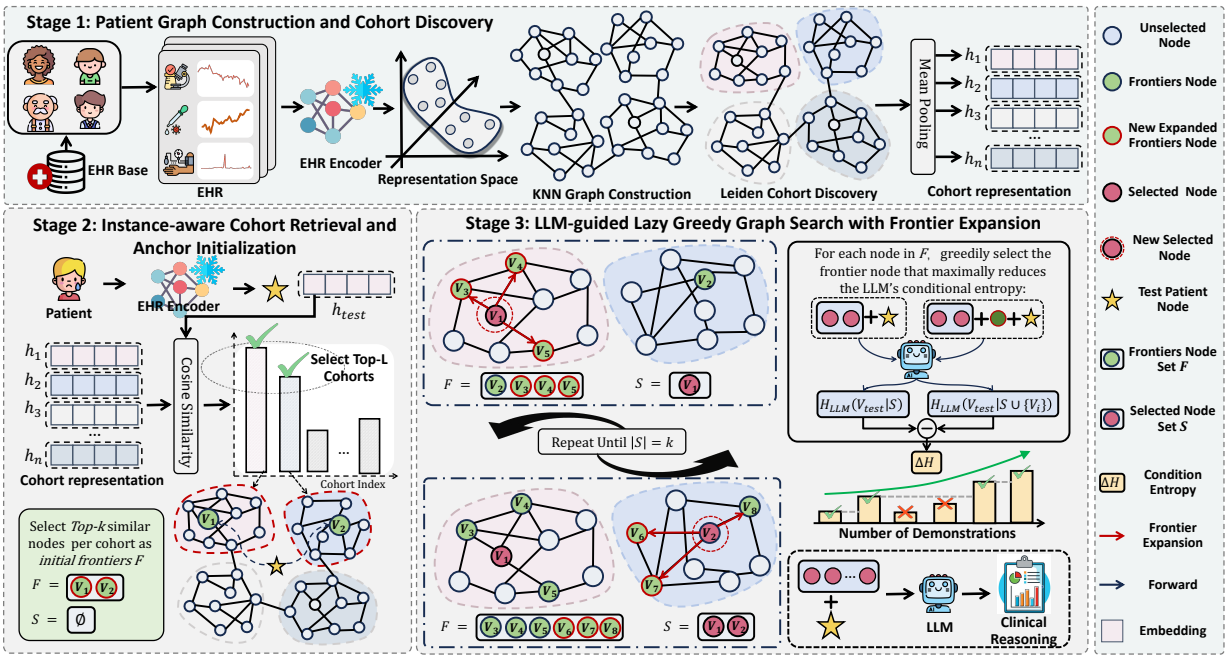

Figure 5: Illustration of *GraphWalker*.

3.2 Instance-aware Cohort Retrieval and Anchor Initialization

Instance-aware Cohort Retrieval. Given a target patient with EHR record $\mathbf{X}^{(q)}$, we obtain its embedding $\mathbf{h}^{(q)} = \mathcal{M}_{\text{exp}}(\mathbf{X}^{(q)})$ using the same pretrained EHR encoder as in Stage 1. To account for cohort awareness and avoid reasoning over isolated patient instances, we retrieve clinically relevant cohorts by aligning $\mathbf{h}^{(q)}$ with cohort prototypes, thereby grounding the subsequent case search in population-level shared clinical patterns. Specifically, we select the top-K_c cohorts whose prototypes $\mathbf{z}_m$ are most similar to $\mathbf{h}^{(q)}$:

$$\mathcal{C}_{\text{ret}}^{(q)} = \text{Top-}K_c\big(\text{sim}(\mathbf{h}^{(q)}, \mathbf{z}_m)\big). \tag{2}$$

This step situates the target patient within clinically coherent regions of the patient graph, providing a focused context for the subsequent step.

Anchor Initialization. Given the retrieved cohorts $\mathcal{C}_{\text{ret}}^{(q)}$, we initialize the search frontier by selecting anchor nodes within each cohort. Specifically, for each cohort $\mathcal{C}_m \in \mathcal{C}_{\text{ret}}^{(q)}$, we select the top-$K_a$ patients whose embeddings are most similar to $\mathbf{h}^{(q)}$:

$$\mathcal{V}_m^{(q)} = \text{Top-}K_a\big(\text{sim}(\mathbf{h}^{(q)}, \mathbf{h}_i)\big)_{v_i \in \mathcal{C}_m}. \tag{3}$$

The initial search frontier is then defined as the union of anchors across retrieved cohorts: $\mathcal{F}^{(q)} = \bigcup_{\mathcal{C}_m \in \mathcal{C}_{\text{ret}}^{(q)}} \mathcal{V}_m^{(q)}$.

3.3 LLM-guided Lazy Greedy Search with Frontier Expansion

LLM-guided Greedy Selection. Starting from the initial frontier $\mathcal{F}^{(q)}$, we iteratively expand the demonstration set $\mathcal{S}^{(q)}$ by selecting candidates that maximize the marginal *information gain* for the target patient. We extend the entropy-based information gain to a *composition-conditioned* criterion that scores each candidate against the running demonstration set rather than independently, an aspect overlooked by prior work.

Formally, let $H_\theta(x_{\text{test}} \mid \mathcal{S})$ denote the conditional entropy of the LLM's prediction on the target query x_{test} given a demonstration composition $\mathcal{S}$, capturing the model's predictive uncertainty under that composition. For a candidate node $v_i \in \mathcal{F}^{(q)}$, its *marginal information gain* with respect to the current demonstration set $\mathcal{S}^{(q)}$ is defined as

$$\Delta H_\theta(v_i \mid \mathcal{S}^{(q)}) = H_\theta(x_{\text{test}} \mid \mathcal{S}^{(q)}) - H_\theta(x_{\text{test}} \mid \mathcal{S}^{(q)} \cup \{v_i\}). \tag{4}$$

A larger $\Delta H_\theta(\cdot)$ indicates that v_i provides more complementary information that enhances the collective reasoning utility of the current composition. At each step, we select the frontier node with the largest marginal gain and add it to the demonstration set:

$$v^* = \arg\max_{v_i \in \mathcal{F}^{(q)}} \Delta H_\theta\big(v_i \mid \mathcal{S}^{(q)}\big), \quad \mathcal{S}^{(q)} \leftarrow \mathcal{S}^{(q)} \cup \{v^*\}. \tag{5}$$

To avoid recomputing ΔH_θ for every frontier node at each iteration, we implement this procedure with a *lazy greedy* strategy, maintaining a priority queue

with cached marginal gains and recomputing them only when necessary (Minoux, 2005); further details are provided in Appendix G. The updated demonstration set then triggers the graph-based frontier expansion step.

Graph-based Frontier Expansion. After each selection step, we expand the search frontier to introduce new candidates that are conditionally relevant to the current demonstration composition. Specifically, given the newly selected node v^*, we add its graph neighbors to the frontier while excluding previously selected nodes:

$$\mathcal{F}^{(q)} \leftarrow \left(\mathcal{F}^{(q)} \cup \mathcal{N}(v^*)\right) \setminus \mathcal{S}^{(q)}, \tag{6}$$

where $\mathcal{N}(v^*) = \{v_j \mid (v^*, v_j) \in \mathcal{E}\}$ denotes the neighbors of v^* in the patient graph. This expansion exploits the local connectivity of the patient graph, where neighboring nodes are expected to share locally coherent clinical patterns. Conditioned on the fact that v^* yields high information gain for the target query, its neighbors are more likely to provide complementary clinical evidence that further enhances the reasoning utility of the current demonstration composition.

Stopping Criterion. The procedure terminates when either (i) a predefined budget of K demonstrations is reached, or (ii) no candidate in the current frontier yields a positive marginal information gain, preventing the inclusion of misleading or uninformative demonstrations.

LLM Inference with Selected Demonstrations. After selection, the demonstration set $\mathcal{S}^{(q)}$ is used to construct a few-shot prompt (see Appendix H for details), which is then fed to the target LLM to generate the final prediction. The full algorithm is provided in Appendix F.

4 Experiment

In this section, we conduct extensive experiments to answer: (**RQ1**) Does *GraphWalker* outperform state-of-the-art demonstration selection baselines for clinical EHR reasoning? (**RQ2**) How sensitive is *GraphWalker* to its key components and hyper-parameters, and what is its runtime efficiency? (**RQ3**) How does parameter-free *GraphWalker* compare against supervised EHR predictive models, both in-domain and under cross-dataset distribution shift? (**RQ4**) Can *GraphWalker* serve as a composable patient-analogy skill within LLM-based agentic clinical workflows?

4.1 Experimental Setup

Datasets and Tasks. We evaluate *GraphWalker* on two widely-used EHR benchmarks: **MIMIC-III** (Johnson et al., 2016) and **MIMIC-IV** (Johnson et al., 2023), across three standard clinical prediction tasks: in-hospital mortality, length-of-stay (LOS), and ICU readmission. Dataset statistics, task definitions, and preprocessing details are in Appendix B.1 and B.2.

Baselines. We compare *GraphWalker* against representative demonstration selection baselines from two perspectives. (1) Data perspective: *Semantic-emb* (Zhang et al., 2025b), *EHR-emb* (Yu et al., 2024), and *Time-series* (Kawarada et al., 2024). (2) Model perspective: *SPELL* (Gonen et al., 2023), *Influence* (Nguyen and Wong, 2023), *IDS* (Qin et al., 2024), *CONE* (Peng et al., 2024), *LMS3* (Liu et al.), *GradSel* (Zhang et al., 2025c), and *Delta-KNN* (Li et al., 2025). We also include *Zero-shot* and *Random* as reference baselines. Implementation details of baselines and ours are in Appendix B.3 and B.4.

Metrics. We report AUROC, AUPRC, and F1-score for mortality and readmission; following Harutyunyan et al. (2019), we report macro- and micro-averaged ROC for LOS.

4.2 Main Results (RQ1)

We report the main results with *Qwen3-14B* and *LLaMA-3.1-8B-Instruct* in Table 1, and defer the full results with the larger *Qwen3-32B* backbone to Appendix Table 3. We summarize the key observations as follows:

Takeaway ❶: *GraphWalker* consistently achieves the strongest performance across backbones and tasks. Over the second-best baseline, *GraphWalker* attains an average absolute improvement of +9.93↑ in AUROC, +12.99↑ in AUPRC, +7.89↑ in F1, +2.86↑ in ma-ROC, and +4.31↑ in mi-ROC.

Takeaway ❷: Dual-perspective selection outperforms all single-perspective baselines. *GraphWalker* consistently outperforms existing *single-perspective* methods, empirically validating the advantage of a dual-view demonstration selection paradigm that bridges clinical similarity and LLM-internal reasoning dynamics.

4.3 Framework and Efficiency Analysis (RQ2)

Component Ablation. We ablate each component of *GraphWalker* on two EHR benchmarks using *Qwen3-14B*; results are reported in Table 2. The three ablations directly verify the three challenges identified in Section 2. *(i) w/o EHR-emb*

Method		MIMIC-III Mortality			MIMIC-III LOS		MIMIC-IV Readmission		
Paradigm	**Approach**	AUROC ↑	AUPRC ↑	F1 ↑	ma-ROC ↑	mi-ROC ↑	AUROC ↑	AUPRC ↑	F1 ↑
Qwen3-14B									
Vanilla	Zero-shot	64.09±5.26	17.58±3.98	31.62±5.44	65.22±3.00	37.49±2.48	41.11±4.77	27.11±4.51	41.66±3.80
	Random	66.33±5.12	31.04±6.89	43.00±5.88	58.12±1.98	61.28±3.35	49.98±4.47	24.81±3.63	40.77±4.05
Data Perspective	Semantic-emb	64.43±6.71	21.78±6.09	33.60±6.16	57.06±3.77	66.60±2.60	45.12±4.13	26.91±4.99	42.01±3.94
	EHR-emb	75.24±5.86	29.52±8.20	42.49±8.88	62.03±3.13	66.98±2.53	57.79±4.37	32.73±5.01	44.87±4.01
	Time-series	62.28±5.09	29.62±6.10	39.34±6.13	58.03±3.50	55.15±2.62	53.46±4.70	26.97±4.05	40.55±3.91
Model Perspective	SPELL	75.37±4.83	32.23±6.55	50.32±6.86	63.73±3.65	75.78±2.18	48.82±4.37	24.37±3.38	40.03±3.98
	Influence	60.54±5.64	27.20±6.59	36.10±6.88	50.03±2.64	62.70±2.30	52.85±3.28	28.55±3.75	43.19±3.85
	IDS	59.14±4.53	19.73±3.93	32.97±4.45	49.86±2.06	37.41±1.11	47.41±4.51	25.43±3.48	42.68±3.79
	CONE	64.69±4.74	27.90±5.44	38.82±5.01	57.84±3.66	67.91±2.57	45.76±4.72	27.75±4.60	41.89±3.79
	LMS3	66.54±5.61	26.92±6.55	39.07±7.46	59.44±3.21	31.89±2.54	44.34±4.57	24.68±3.47	42.28±3.81
	Delta-KNN	51.52±6.34	15.66±4.31	25.22±4.87	59.04±3.61	63.43±2.51	47.11±4.81	24.41±3.90	40.00±3.91
	GradSel	62.09±5.40	32.20±7.03	36.68±5.68	66.80±3.37	73.57±2.01	45.73±4.29	25.39±3.44	41.83±3.81
Ours	*GraphWalker*	**84.19±3.35**	**59.46±7.64**	**58.24±5.89**	**69.51±3.16**	**84.64±2.14**	**75.91±4.11**	**60.61±6.19**	**58.01±5.33**
LLaMA-3.1-8B-Instruct									
Vanilla	Zero-shot	65.35±4.90	21.72±4.61	35.75±5.22	43.16±4.25	74.13±2.63	54.40±5.02	29.88±4.99	41.24±4.14
	Random	52.04±5.55	18.81±4.78	28.90±4.11	45.66±4.24	80.14±2.31	48.63±5.03	27.06±4.68	39.85±3.85
Data Perspective	Semantic-emb	56.90±5.44	22.32±4.69	35.15±5.10	53.78±3.87	81.49±2.26	43.55±4.33	23.69±3.25	42.37±3.83
	EHR-emb	65.78±4.24	24.30±4.39	41.89±5.14	45.72±4.08	74.78±2.63	63.75±4.78	46.32±6.37	49.81±4.88
	Time-series	56.38±6.06	24.86±5.56	37.16±5.78	46.23±3.73	71.99±2.57	53.45±4.51	27.67±4.43	42.28±4.28
Model Perspective	SPELL	55.43±4.83	20.92±4.44	34.51±4.45	49.48±4.07	82.02±2.24	48.11±4.67	25.03±4.09	40.09±3.86
	Influence	61.73±5.31	31.80±6.55	38.39±5.19	55.57±4.39	82.63±2.37	49.64±3.99	28.40±4.17	42.65±3.81
	IDS	59.89±5.52	28.40±6.61	36.11±5.13	42.12±3.58	71.14±2.10	45.25±4.52	24.86±3.59	42.65±3.81
	CONE	50.40±5.29	18.93±3.97	32.79±4.54	38.31±3.82	76.94±2.53	51.67±4.73	29.42±4.45	42.73±3.86
	LMS3	51.80±5.73	16.41±3.38	29.37±4.67	39.60±3.86	74.05±2.68	54.77±4.81	30.95±4.55	45.41±4.73
	Delta-KNN	57.42±4.91	21.48±4.12	33.61±4.43	47.69±3.82	75.71±2.60	51.23±4.62	26.50±4.15	41.13±4.00
	GradSel	47.26±5.37	17.32±3.22	31.23±4.04	53.92±4.14	80.83±2.29	52.33±4.04	28.70±4.02	42.17±3.83
Ours	*GraphWalker*	**76.71±3.58**	**35.09±6.40**	**50.13±5.52**	**57.06±3.98**	**83.55±2.16**	**68.32±4.19**	**46.37±6.65**	**51.54±4.45**

Table 1: Performance comparison on MIMIC-III and MIMIC-IV datasets using *Qwen3-14B* and *Llama3.1-8B-Instruct* as backbone LLMs. All metrics are higher-is-better (↑); values are mean±std over three random seeds. **Bold**: best; underlined: second-best.

Method	MIMIC-III Mortality			MIMIC-III LOS		MIMIC-IV Readmission		
	AUROC ↑	AUPRC ↑	F1 ↑	ma-ROC ↑	mi-ROC ↑	AUROC ↑	AUPRC ↑	F1 ↑
GraphWalker	**84.19±3.35**	**59.46±7.64**	**58.24±5.89**	**69.51±3.16**	**84.64±2.14**	**75.91±4.11**	**60.61±6.19**	**58.01±5.33**
w/o EHR-emb	61.09±5.32	31.13±6.81	37.50±5.73	62.03±3.40	72.63±2.46	45.30±4.55	26.69±4.20	41.74±3.79
w/o Cohort	81.19±3.78	48.27±8.01	55.06±6.18	65.85±3.75	81.16±2.27	75.65±4.04	59.61±6.09	56.87±5.19
w/o Greedy & Frontier Expansion	69.43±5.80	46.72±8.24	55.34±7.22	60.93±3.30	53.33±2.52	62.43±4.72	43.69±6.25	47.24±4.90
w/o Early Stop	81.34±3.74	48.66±7.91	55.99±6.18	66.31±3.23	82.77±2.14	75.53±4.09	59.13±6.12	57.38±5.22
Leiden → Louvain	83.15±4.84	57.75±9.06	56.37±8.02	68.16±3.30	83.04±2.06	75.68±4.02	59.83±6.18	55.97±4.81

Table 2: Ablation study of *GraphWalker* using *Qwen3-14B* as backbone LLM on MIMIC-III and MIMIC-IV datasets. All evaluation metrics are defined such that higher values indicate better performance (↑). The best results are highlighted in **Bold**.

(−18.50 F1): **generic semantic similarity is insufficient.** Replacing the EHR encoder with *Qwen3-Embedding-8B* fails to capture clinically meaningful relationships required for effective graph-based search (*Observation I*). *(ii) w/o Cohort Discovery* (−6.09 AUPRC): **cohort-level retrieval matters.** Directly initializing anchors as the top-K_a nearest patients reduces *GraphWalker* to instance-level retrieval, which is vulnerable to noise from idiosyncratic cases (*Observation II*). Replacing Leiden with Louvain (Blondel et al., 2008) causes consistent drops, indicating that higher-quality partitions further benefit cohort retrieval and frontier expansion. *(iii) w/o Greedy & Frontier Expansion* (−14.83 AUPRC): **independent scoring fails.** Selecting demonstrations solely by individual ΔH scores implicitly assumes linear utility accumulation, ignoring redundancy and interaction effects that cause diminishing marginal gains (*Observation III*). *w/o Early Stop* further degrades performance by admitting demonstrations with negative marginal gains. *GraphWalker* further generalizes across multiple pretrained EHR encoders, confirming its effectiveness is not tied to any specific representation (Appendix C.2.1, Table 4).

Hyper-parameter Sensitivity and Shot Scaling Analysis. For each of the three key hyper-parameters (cohort count K_c, anchor count K_a, and graph neighborhood size k_g), *GraphWalker* performs best at intermediate values and degrades when the parameter is set too small or too large (Appendix Figure 6), reflecting an interpretable

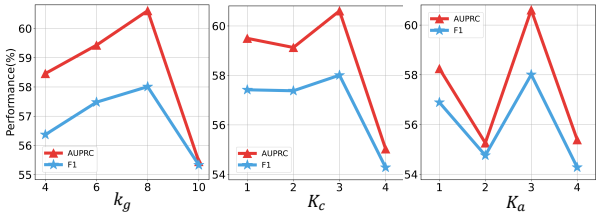

Figure 6: Performance of *GraphWalker* over different hyper-parameters on MIMIC-IV.

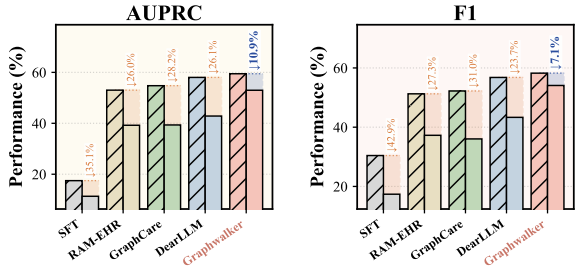

Figure 7: In-domain (hatched) vs. cross-dataset OOD (solid) on MIMIC-III mortality. Left: AUPRC; right: F1. Shaded regions denote the relative drop. AUROC: Appendix C.3.

coverage-noise trade-off rather than brittle sensitivity. Under varying demonstration budgets K, ***GraphWalker* is the only method whose performance consistently improves with more demonstrations** (Figures 8 and 9), uniquely sustaining a stable ΔH increase across shots (Figure 4) by composing demonstrations for joint information gain rather than scoring them independently. Baselines, in contrast, exhibit non-monotonic trends, confirming that the gain comes from *how* demonstrations are selected and composed, not from *how many*. Detailed analyses are in Appendix C.2.2 and C.2.3.

Cost Analysis. *GraphWalker* incurs only **seconds-level additional inference overhead** per patient, while delivering substantial performance gains (Table 1). This favorable performance–latency trade-off is enabled by lazy greedy optimization with frontier expansion; the detailed runtime breakdown is provided in Appendix C.2.4.

4.4 Compare to Supervised Methods (RQ3)

We evaluate *GraphWalker* against representative supervised models on MIMIC-III mortality in two regimes: **(i) in-domain** (trained and tested on MIMIC-III), and **(ii) cross-dataset OOD** (trained on MIMIC-IV, tested on MIMIC-III without fine-tuning; *GraphWalker*'s graph and retrieval base are built exclusively on MIMIC-IV's training split). See Figure 7 and Appendix C.3 for details.

Takeaway ❸: In-domain, *GraphWalker* outperforms all four supervised baselines, including LLM-augmented hybrid pipelines such as RAM-EHR (Xu et al., 2024a), GraphCare (Jiang et al., 2024), and DearLLM (Xu et al., 2025). Under cross-dataset distribution shift, supervised methods degrade notably (15.4%–42.9% drop), whereas *GraphWalker* stays stable (3.4%–10.9% drop) and surpasses all baselines, reflecting a paradigm-level shift from source-specific feature correlations to reasoning by analogy over retrieved patient cases.

4.5 Agent Skill Integration (RQ4)

Beyond standalone demonstration selection, we examine whether *GraphWalker* can serve as a **composable patient-analogy skill**: a pluggable upstream module that supplies high-quality patient case context to downstream agentic workflows. To this end, we (i) implement a **black-box variant** of *GraphWalker* that replaces entropy-based scoring with LLM self-evaluation, applicable to closed-source LLMs without access to logits; and (ii) integrate it with **ReAct** (Yao et al., 2022), where the selected demonstrations remain unchanged but direct answer generation is replaced with ReAct-style reasoning. Experiments use MIMIC-III mortality; details in Appendix C.4.

Takeaway ❹: On closed-source LLMs, the black-box variant of *GraphWalker* substantially outperforms random and similarity-based baselines, confirming that its advantage stems from cohort-guided, target-conditioned demonstration selection rather than any specific scoring interface. When combined with ReAct, *GraphWalker* further improves across all three metrics whereas ReAct brings limited gains to weaker baselines, positioning *GraphWalker* as a composable patient-analogy skill that complements agent-style reasoning.

Extended Investigations and Key Insights. We further evaluate *GraphWalker* on additional medical reasoning benchmarks spanning diverse task formats, where it achieves an average gain of 3.5%↑ (Table 8; Appendix C.5), demonstrating its **generalizability to broader real-world medical reasoning scenarios**. A **case study** is also provided in Appendix D.

5 Conclusion

We presented *GraphWalker*, an inference-time framework for clinical reasoning by patient analogy that addresses perspective limitation, cohort awareness, and information aggregation through dual-view selection, cohort discovery, and lazy greedy search with frontier expansion. Experiments across EHR and medical reasoning benchmarks demonstrate its effectiveness, positioning *GraphWalker* as a pluggable patient-analogy skill for LLM-based clinical reasoning.

6 Limitations

Although *GraphWalker* shows consistent gains across EHR prediction, additional medical reasoning benchmarks, and black-box or agentic extensions, several limitations remain. Our current evaluation still focuses on structured EHR-based prediction and selected medical reasoning settings, partly due to computational constraints. Extending *GraphWalker* to broader clinical scenarios, such as multimodal records and interactive clinical workflows, is an especially promising direction for future work, and we expect its core cohort-guided and target-conditioned demonstration selection mechanism to remain effective in such richer clinical environments. Moreover, *GraphWalker* is not restricted to the white-box few-shot setting: it can already be adapted to black-box LLMs and integrated with agentic reasoning scaffolds. While we have not yet systematically evaluated it in more complex clinical agent systems involving iterative tool use and evolving reasoning skills, this remains a particularly promising direction for future work, and our current black-box and ReAct-based results already provide encouraging preliminary evidence.

7 Ethical Considerations

Clinical decision-making is a complex and high-stakes process that relies on comprehensive medical evidence and professional judgment. Although LLMs can assist clinicians by organizing information, highlighting relevant patterns, or supporting exploratory reasoning, there is a risk that their outputs may be misunderstood or inappropriately treated as definitive medical advice. We emphasize that the system studied in this work is not intended to function as a diagnostic tool, nor to replace qualified healthcare professionals. Any model-generated reasoning or suggestions should be interpreted only as auxiliary information and must be considered alongside established clinical records, examinations, and expert assessment. All experiments in this study are conducted on publicly available and de-identified datasets, and no personally identifiable information or human subject data is involved. We acknowledge that the underlying datasets may exhibit demographic imbalances, including disparities across race, ethnicity, age, and other population attributes, which could potentially affect model behavior and generalizability. Our method is developed and evaluated solely for research purposes and is not deployed in real-world clinical settings. While the proposed approach improves the reliability and coherence of in-context reasoning under limited information, any future clinical application would require extensive validation, careful safety evaluation, and strict compliance with medical regulations and ethical guidelines. We view this work as a step toward more responsible and transparent medical AI systems, rather than a substitute for human clinical expertise.

References

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Katherine E Brown, Chao Yan, Zhuohang Li, Xinmeng Zhang, Benjamin X Collins, You Chen, Ellen Wright Clayton, Murat Kantarcioglu, Yevgeniy Vorobeychik, and Bradley A Malin. 2024. Not the models you are looking for: Traditional ml outperforms llms in clinical prediction tasks. *medRxiv*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Canyu Chen, Jian Yu, Shan Chen, Che Liu, Zhongwei Wan, Danielle Bitterman, Fei Wang, and Kai Shu. 2024. Clinicalbench: Can llms beat traditional ml models in clinical prediction? *arXiv preprint arXiv:2411.06469*.

Hongxin Ding, Yue Fang, Runchuan Zhu, Xinke Jiang, Jinyang Zhang, Yongxin Xu, Weibin Liao, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025a. 3ds: Medical domain adaptation of llms via decomposed difficulty-based data selection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19473–19495.

Hongxin Ding, Baixiang Huang, Yue Fang, Weibin Liao, Xinke Jiang, Zheng Li, Junfeng Zhao, and Yasha Wang. 2025b. Promed: Shapley information gain guided reinforcement learning for proactive medical llms. *arXiv preprint arXiv:2508.13514*.

Yue Fang, Yuxin Guo, Jiaran Gao, Hongxin Ding, Xinke Jiang, Weibin Liao, Yongxin Xu, Yinghao Zhu, Zhibang Yang, Liantao Ma, and 1 others. 2025. Toward better ehr reasoning in llms: Reinforcement learning with expert attention guidance. *arXiv preprint arXiv:2508.13579*.

Yue Fang, Yongxin Xu, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Hongyan Jin. 2023. A method and

practice for menopausal disease prediction based on knowledge graph. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pages 10–18. IEEE.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. 2025. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*.

Junyi Gao, Yinghao Zhu, Wenqing Wang, Zixiang Wang, Guiying Dong, Wen Tang, Hao Wang, Yasha Wang, Ewen M Harrison, and Liantao Ma. 2024. A comprehensive benchmark for covid-19 predictive modeling using electronic health records in intensive care. *Patterns*, 5(4).

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):96.

SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and 1 others. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Pengcheng Jiang, Cao Danica Xiao, Adam Cross, and Jimeng Sun. 2024. Graphcare: Enhancing healthcare predictions with personalized knowledge graphs. In *International Conference on Learning Representations*, volume 2024, pages 16839–16866.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, and 1 others. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Sci. Data*.

Masayuki Kawarada, Tatsuya Ishigaki, Goran Topić, and Hiroya Takamura. 2024. Demonstration selection strategies for numerical time series data-to-text. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7378–7392.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Chuyuan Li, Raymond Li, Thalia S Field, and Giuseppe Carenini. 2025. Delta-knn: Improving demonstration selection in in-context learning for alzheimer's disease detection. *arXiv preprint arXiv:2506.03476*.

Weibin Liao, Yinghao Zhu, Zhongji Zhang, Yuhang Wang, Zixiang Wang, Xu Chu, Yasha Wang, and Liantao Ma. 2025. Learnable prompt as pseudo-imputation: Rethinking the necessity of traditional ehr data imputation in downstream clinical prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, page 765–776.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pages 100–114.

Jiayu Liu, Zhenya Huang, Chaokun Wang, Xunpeng Huang, ChengXiang Zhai, and Enhong Chen. What makes in-context learning effective for mathematical reasoning. In *Forty-second International Conference on Machine Learning*.

Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020a. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 825–832.

Liantao Ma, Chaohe Zhang, Junyi Gao, Xianfeng Jiao, Zhihao Yu, Yinghao Zhu, Tianlong Wang, Xinyu Ma, Yasha Wang, Wen Tang, and 1 others. 2023. Mortality prediction with adaptive feature importance recalibration for peritoneal dialysis patients. *Patterns*, 4(12).

Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. 2020b. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 833–840.

Michel Minoux. 2005. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*, pages 234–243. Springer.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

Geoffrey Norman. 2006. Building on experience—the development of clinical reasoning.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. In *ACL (1)*.

Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. 2024. In-context learning with iterative demonstration selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7441–7455.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Olle Ten Cate, Eugène JFM Custers, and Steven J Durning. 2017. Principles and practice of case-based clinical reasoning education: a method for preclinical students.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Xidong Wang, Guiming Chen, Song Dingjie, Zhang Zhiyi, Zhihong Chen, Qingying Xiao, Junying Chen, Feng Jiang, Jianquan Li, Xiang Wan, and 1 others. 2024. Cmb: A comprehensive medical benchmark in chinese. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6184–6205.

Brian J Wells, Kevin M Chagin, Amy S Nowacki, and Michael W Kattan. 2013. Strategies for handling missing data in electronic health record derived data. *Egems*, 1(3):1035.

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Bowen Jin, May Dongmei Wang, Joyce Ho, and Carl Yang. 2024a. Ram-ehr: Retrieval augmentation meets clinical predictions on electronic health records. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 754–765.

Yongxin Xu, Xu Chu, Kai Yang, Zhiyuan Wang, Peinie Zou, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2023a. Seqcare: Sequential training with external medical knowledge graph for diagnosis prediction in healthcare data. In *Proceedings of the ACM Web Conference 2023*, pages 2819–2830.

Yongxin Xu, Xinke Jiang, Xu Chu, Rihong Qiu, Yujie Feng, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2025. Dearllm: Enhancing personalized healthcare via large language models-deduced feature correlations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 941–949.

Yongxin Xu, Xinke Jiang, Xu Chu, Yuzhen Xiao, Chaohe Zhang, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2024b. Protomix: Augmenting health status representation learning via prototype-based mixup. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3633–3644.

Yongxin Xu, Kai Yang, Chaohe Zhang, Peinie Zou, Zhiyuan Wang, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2023b. Vecocare: Visit sequences-clinical notes joint learning for diagnosis prediction in healthcare data. In *IJCAI*, volume 23, pages 4921–4929.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Zhihao Yu, Chu Xu, Yujie Jin, Yasha Wang, and Junfeng Zhao. 2024. Smart: Towards pre-trained missing-aware model for patient health status prediction. *Advances in Neural Information Processing Systems*, 37:63986–64009.

Jinyang Zhang, Yue Fang, Hongxin Ding, Weibin Liao, Muyang Ye, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025a. Adept: Continual pretraining via adaptive expansion and dynamic decoupled tuning. *arXiv preprint arXiv:2510.10071*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Ziniu Zhang, Zhenshuo Zhang, Dongyue Li, Lu Wang, Jennifer Dy, and Hongyang R Zhang. 2025c. Linear-time demonstration selection for in-context learning via gradient estimation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16470–16488.

Yinghao Zhu, Junyi Gao, Zixiang Wang, Weibin Liao, Xiaochen Zheng, Lifang Liang, Miguel O Bernabeu, Yasha Wang, Lequan Yu, Chengwei Pan, and 1 others. 2024a. Clinicrealm: Re-evaluating large language models with conventional machine learning for non-generative clinical prediction tasks. *arXiv preprint arXiv:2407.18525*.

Yinghao Zhu, Junyi Gao, Zixiang Wang, Weibin Liao, Xiaochen Zheng, Lifang Liang, Miguel O Bernabeu, Yasha Wang, Lequan Yu, Chengwei Pan, and 1 others. 2026. Clinicrealm: Re-evaluating large language models with conventional machine learning for non-generative clinical prediction tasks. *npj Digital Medicine*.

Yinghao Zhu, Wenqing Wang, Junyi Gao, and Liantao Ma. 2024b. Pyehr: A predictive modeling toolkit for electronic health records.

A Related Work

Recent years have witnessed growing interest in applying LLMs to clinical reasoning over EHRs. One line of work adapts LLMs to the medical domain through continual pre-training or post-training on clinical corpora (Zhang et al., 2025a; Ding et al., 2025a; Fang et al., 2025; Ding et al., 2025b); a parallel line builds hybrid pipelines that distill LLM-derived knowledge into supervised EHR predictors (Jiang et al., 2024; Xu et al., 2025, 2024a). Both directions, however, rely on parameter updates over task-specific labeled data. The former leaves LLMs limited in modeling longitudinal EHR data and activating clinically meaningful temporal patterns stored in parametric memory (Brown et al., 2024; Chen et al., 2024; Zhu et al., 2024a); the latter inherits the brittleness of its downstream supervised predictor to variations in feature order, availability, and encoding across heterogeneous EHR systems (Fang et al., 2025). These limitations motivate the exploration of inference-time adaptation strategies.

Demonstration Selection for ICL. In-context learning (ICL) (Brown et al., 2020) enables LLMs to adapt at inference time by conditioning on a small set of demonstrations, and has been widely studied in both general NLP and domain-specific settings. Existing demonstration selection methods can be broadly grouped into two paradigms. Data-perspective approaches retrieve demonstrations based on embedding similarity, assuming that representation-level proximity correlates with reasoning utility (Liu et al., 2022; Hongjin et al., 2022; Robertson et al., 2009; Yu et al., 2024). Model-perspective approaches instead leverage LLM-internal signals, such as perplexity, conditional entropy, gradients, or influence estimates, to assess the usefulness of individual demonstrations (Gonen et al., 2023; Peng et al., 2024; Liu et al.; Li et al., 2025; Zhang et al., 2025c; Qin et al., 2024). However, existing demonstration selection methods largely lack a principled framework that jointly incorporates clinical knowledge and model-aware reasoning signals for EHR-based clinical reasoning.

B Experimental Details

B.1 Dataset Descriptions

In this section, we provide a detailed introduction of the datasets used in our study, along with the

preprocessing procedures applied to the structured EHR data.

We conduct experiments on two real-world EHR benchmarks: **MIMIC-III v1.4** (Johnson et al., 2016) and **MIMIC-IV v2.2** (Johnson et al., 2023). Both are large-scale, publicly available critical-care databases collected at the Beth Israel Deaconess Medical Center, providing longitudinal, de-identified ICU records with heterogeneous clinical events (e.g., vital signs, laboratory tests, and medication administrations). The resulting irregular, long-horizon patient trajectories make these datasets well-suited for evaluating clinical reasoning over EHRs.

Data Preprocessing. We preprocess both datasets following the standardized PyEHR (Zhu et al., 2024b) pipeline, which is also adopted by recent EHR benchmarks (Gao et al., 2024). Time-stamped clinical events within each ICU stay are temporally aggregated into daily records; for long stays, we retain the most recent seven days, which prior work has shown to capture the most clinically salient trajectories. Missing values are handled using the Last Observation Carried Forward (LOCF) strategy (Wells et al., 2013).

For each patient, we use *all historical visits except the last one* as input and predict the outcome associated with the last visit. This setup strictly excludes the outcome visit from the input, ensuring that no information from the prediction target is leaked into the input features. Specifically, for **in-hospital mortality**, we predict whether the patient dies during the last visit; for **length-of-stay (LOS)**, we predict the LOS bin (Harutyunyan et al., 2019) of the last visit; for **ICU readmission**, we predict whether a new ICU admission occurs within 30 days of the last visit's discharge. The same prediction protocol is applied uniformly to all baselines and to *GraphWalker*.

To enable controlled comparison across a broad set of LLM backbones and case selection baselines, each requiring inference-time forward passes per patient, we follow prior EHR studies (Zhu et al., 2026) and split each dataset into training, validation, and test sets at an 8:1:1 ratio, with the test set fixed across all experiments to ensure fair comparison.

B.2 Tasks

Definition 1 (EHR Dataset). A patient's EHR is represented as a sequence of T time-ordered visits $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$, where each visit $\mathbf{x}_t = \{l_{t,1}, l_{t,2}, \cdots, l_{t,n_t}\}$ contains n_t lab test features.

Definition 2 (Mortality Prediction). Given $\mathbf{X}$, predict whether the patient will survive the hospital stay. The label $y \in \{0, 1\}$ denotes death ($y = 1$) or survival ($y = 0$).

Definition 3 (Readmission Prediction). Given $\mathbf{X}$, predict whether the patient will be readmitted within 30 days after discharge. The label $y \in \{0, 1\}$ indicates readmission ($y = 1$) or not ($y = 0$).

Definition 4 (Length-of-Stay Prediction). Given $\mathbf{X}$, predict the length of the patient's hospital stay. Following prior work (Harutyunyan et al., 2019), we formulate this task as a classification problem, where the label $y \in \{1, 2, \cdots, C\}$ indicates the length-of-stay category corresponding to predefined time intervals.

B.3 Baseline Implementation Details

We compare *GraphWalker* with representative baselines spanning multiple paradigms. In this subsection, we provide implementation details for each baseline. For all methods, we strictly constrain the LLM to directly output the final prediction in the task-specific format, without generating any intermediate reasoning or explanations, following the prompt templates described in Appendix H.

- **Vanilla ICL baselines.** *Zero-shot* corresponds to a special case of ICL where no demonstration examples are provided, and the LLM directly produces predictions based on the task description and patient EHR data. *Random* samples a fixed number of patient examples uniformly at random as demonstrations. Both baselines adopt the same prompt structure as described in Appendix H, differing only in whether in-context examples are included.
- **Supervised.** *SFT* adapts LLMs to downstream tasks by fine-tuning model parameters on task-specific data. Specifically, we organize the training set into question–answer pairs using the same prompt templates described in Appendix H, and apply parameter-efficient fine-tuning via LoRA (Hu et al., 2022) to update the backbone LLM.
- **Data Perspective Selection.** *Semantic-emb* retrieves top-k demonstrations based on similarity in the embedding space of *Qwen3-Embedding-8B* (Zhang et al., 2025b), a widely used and strong general-purpose semantic model.

For efficiency, we compute embeddings using vLLM (Kwon et al., 2023) to accelerate large-scale embedding inference. *EHR-emb* retrieves top-k demonstrations based on similarity computed from embeddings generated by *SMART* (Yu et al., 2024), a pretrained model specifically designed for EHR data. We follow the hyperparameter settings recommended in the original work and use the `[CLS]` token embedding as the patient representation, as it encodes global semantic information of the longitudinal EHR sequence. *Time-series* (Kawarada et al., 2024) retrieves top-k demonstrations based on the similarity between the test patient and training examples' longitudinal EHR time series. We use a scale-invariant masked cosine similarity by aligning overlapping prefixes, truncating to the minimum length, and computing cosine similarity only over positions with valid values, ignoring missing entries. Demonstrations are ranked by similarity, and the top-k are selected as in-context examples.

- **Model Perspective Selection.** *SPELL* (Gonen et al., 2023) selects demonstrations by ranking candidates with LM perplexity (i.e., teacher-forced next-token cross-entropy) and choosing the lowest-perplexity top-k examples. In our implementation, we score all training examples with the same causal LM and select the top-k demonstrations by sorting the average loss in ascending order. The resulting few-shot set is shared across all test instances and inserted into the ICL prompt as demonstrations, without per-test-instance re-selection.

  Influence (Nguyen and Wong, 2023) estimates the importance of demonstrations by measuring their marginal contribution to the model's performance on a validation set. It calculates influence scores based on the average performance difference between sampled subsets that include a specific example and those that exclude it.

  IDS (Qin et al., 2024) selects demonstrations by calculating the semantic similarity between training examples and reasoning paths generated through Zero-shot CoT. The framework iteratively refines the demonstration set by utilizing the model's self-generated reasoning from the previous round as an updated query for subsequent retrieval.

  CONE (Peng et al., 2024) selects demonstrations by minimizing the conditional entropy of the test input on a per-example basis, without modeling complementarity among selected examples. Following the original implementation, we first perform a semantic pre-filtering step using *Qwen3-Embedding-8B* to retrieve the top-10 semantically similar candidates. We then select the top-k demonstrations with the largest reduction in conditional entropy, where the entropy is computed independently for each candidate example.

  LMS3 (Liu et al.) selects demonstrations by balancing LLM-oriented semantic similarity and inference stability, derived from a theoretical bound on reasoning efficacy. It uniquely incorporates a demonstration rejection mechanism that adaptively filters out unsuitable examples to prevent negative transfer.

  Delta-KNN (Li et al., 2025) selects demonstrations by prioritizing candidates that were empirically observed to be most helpful for similar inputs, using a cached example-to-example gain matrix. In our implementation, we first take a fixed training subset of examples to construct the Delta matrix via paired zero-shot vs. one-shot inference, and cache the resulting matrix for each dataset/LLM setting. At test time, we use *Qwen3-Embedding-8B* to retrieve the top-10 semantically similar candidates, then re-rank them with the Delta-based scores and select the final top-k demonstrations. The selected three demonstrations are re-selected per test instance and inserted into the ICL prompt as demonstrations for final inference.

  GradSel (Zhang et al., 2025c) estimates the utility of sampled demonstration subsets using a first-order approximation in the input-embedding space, based on the model output and gradients computed at anchor prompts. It then aggregates subset-level scores into point-wise demonstration scores and selects the top-ranked examples as final demonstrations.

B.4 Implementation Details of *GraphWalker*

For patient graph construction, we set the number of nearest neighbors to $k_g = 8$. For instance-aware cohort retrieval, the number of retrieved cohorts is set to $K_c = 3$, and within each retrieved cohort, we select $K_a = 3$ anchor nodes to initialize the search frontier. A detailed analysis and discussion of these key hyperparameters are provided in Appendix C.2.2. For graph-based cohort discovery, the resolution parameter of the Leiden algorithm is set to 0.9. All experiments are repeated three times with different random seeds, and we report

the mean performance along with the standard deviation. To accelerate the computation of conditional entropy during LLM-guided demonstration selection, we adopt `vLLM` (Kwon et al., 2023) for efficient inference.

C Additional Experiment Results

C.1 RQ1 Results

To further evaluate the robustness of *GraphWalker* under larger backbone models, we additionally report results using *Qwen3-32B*. As shown in Table 3, *GraphWalker* consistently achieves the best overall performance across all three clinical prediction tasks, including MIMIC-III Mortality, MIMIC-III LOS, and MIMIC-IV Readmission.

C.2 RQ2 Results

C.2.1 Effect of Different EHR Encoders

Setup. To assess the robustness of *GraphWalker* to the choice of EHR representation, we instantiate it with three pretrained EHR encoders of different architectures: **SMART** (Yu et al., 2024), **ConCare** (Ma et al., 2020b), and **AdaCare** (Ma et al., 2020a). For each encoder, we compare two settings: (i) the encoder fine-tuned end-to-end as a task-specific supervised predictor (*encoder baseline*), and (ii) *GraphWalker* using the same pretrained encoder *without fine-tuning* for graph-based demonstration selection. Note that the encoder baseline here differs from *EHR-emb* in Table 1, which uses SMART embeddings for demonstration retrieval without fine-tuning.

Results. Table 4 shows that *GraphWalker* consistently improves over all encoder baselines across both MIMIC-III mortality and MIMIC-IV readmission, with consistent absolute gains across metrics and encoders. The improvements hold across encoders of distinct architectural styles, indicating that *GraphWalker*'s effectiveness arises from its demonstration selection mechanism rather than from any specific EHR representation. Notably, *GraphWalker* surpasses these encoders *even when they are fine-tuned end-to-end on the target task*, underscoring that the gains come from how demonstrations are selected and composed, not merely from the underlying encoder.

C.2.2 Detailed Hyper-Parameter Analysis

We analyze three key hyper-parameters of *GraphWalker* on the validation set of *MIMIC-IV Readmission* with *Qwen3-14B* as the backbone LLM: the graph neighborhood size k_g, the number of retrieved cohorts K_c, and the number of anchor nodes per cohort K_a. All other settings are kept fixed.

Overall, *GraphWalker* performs best under *moderate* settings that balance candidate diversity and clinical coherence. Across all three hyper-parameters, performance remains strongest within a small intermediate range before declining when the candidate space becomes overly sparse or overly noisy, reflecting an interpretable coverage–noise trade-off rather than brittle sensitivity to narrow tuning.

Graph Neighborhood Size k_g. We vary k_g from 4 to 10 (Figure 6(a)). Performance steadily improves as k_g grows from 4 to 8, indicating that a moderate neighborhood captures richer population-level structure. When k_g further increases to 10, performance declines as the graph becomes overly dense and introduces weakly related neighbors, blurring local structure and propagating less informative signals during frontier expansion.

Number of Retrieved Cohorts K_c. Varying K_c from 1 to 4 (Figure 6(b)), both AUPRC and F1 steadily improve from 1 to 3, suggesting that retrieving multiple cohorts enriches the candidate pool with diverse but clinically related subpopulations and mitigates over-specialization to a single cohort. Performance declines at $K_c = 4$, where additional loosely related cohorts begin to dilute clinically meaningful signals.

Anchor Nodes per Cohort K_a. Performance peaks at $K_a = 3$ (Figure 6(c)). A moderate number of anchors balances intra-cohort coverage and candidate quality—sufficient to represent diverse yet clinically consistent profiles within each cohort. Larger K_a over-expands the initial frontier with marginal anchors, increasing redundancy and reducing the effectiveness of interaction-aware demonstration selection.

C.2.3 Shot Scaling Analysis

To examine the effect of shot number on *GraphWalker*, we evaluate its performance under varying k-shot settings ($k \in \{1, 2, 3, 4\}$) on *MIMIC-III Mortality* and *MIMIC-IV Readmission* with *Qwen3-14B* as the backbone LLM (Figures 8 and 9).

***GraphWalker* consistently outperforms all baselines across k-shot settings.** This demonstrates strong robustness to the choice of shot number. In contrast, baseline methods exhibit non-monotonic

Method		MIMIC-III Mortality			MIMIC-III LOS		MIMIC-IV Readmission		
Paradigm	**Approach**	AUROC ↑	AUPRC ↑	F1 ↑	ma-ROC ↑	mi-ROC ↑	AUROC ↑	AUPRC ↑	F1 ↑
Qwen3-32B									
Vanilla	Zero-shot	65.67±5.75	30.54±7.56	38.43±6.99	55.79±3.33	23.53±2.46	44.19±4.57	25.06±3.88	42.57±3.83
	Random	59.90±5.06	24.63±5.29	36.22±4.94	59.50±3.51	80.44±2.00	48.25±4.65	28.13±4.51	42.22±3.77
Data Perspective	Semantic-emb	60.62±5.80	19.87±4.61	32.08±5.59	57.65±3.65	74.16±2.38	47.86±4.71	26.07±3.89	42.47±3.85
	EHR-emb	85.02±3.21	58.56±8.02	58.84±6.01	63.02±3.33	67.00±2.71	61.32±4.45	44.25±5.92	46.24±4.06
	Time-series	63.71±4.93	28.74±6.28	37.58±5.29	51.95±4.04	38.06±2.98	52.34±4.78	27.02±4.39	40.79±3.97
Model Perspective	SPELL	63.26±4.98	27.26±5.64	38.33±5.45	53.87±3.76	75.06±2.27	48.56±4.77	25.90±4.11	40.17±3.86
	Influence	62.45±5.55	29.28±6.41	40.18±6.30	54.24±3.02	41.07±1.56	50.45±3.49	27.63±3.42	42.69±3.94
	IDS	56.72±5.64	28.37±6.91	35.23±5.03	52.47±3.12	45.82±2.47	46.93±4.66	27.71±4.34	43.16±3.79
	CONE	64.28±6.59	27.83±7.35	36.05±6.90	56.30±3.73	68.90±2.64	45.35±4.83	27.85±4.77	41.73±3.79
	LMS3	62.03±5.12	25.14±5.03	36.68±4.79	58.14±3.73	73.15±2.55	45.28±4.40	26.09±4.01	42.79±3.82
	Delta-KNN	61.08±5.83	29.41±6.34	39.36±5.92	56.55±4.09	64.86±2.64	51.75±4.68	26.74±4.26	40.37±3.90
	GradSel	60.77±5.22	24.53±4.77	39.30±5.68	53.87±3.75	44.22±2.55	47.08±4.02	25.53±3.41	41.93±3.80
Ours	*GraphWalker*	**85.88±3.16**	**59.19±8.04**	**61.89±6.07**	**67.40±3.51**	**83.60±2.34**	**77.57±4.01**	**63.09±6.27**	**59.51±4.46**

Table 3: Additional results on MIMIC-III and MIMIC-IV datasets using *Qwen3-32B* as the backbone LLM. All metrics are higher-is-better (↑); values are mean±std over three random seeds. **Bold**: best; underlined: second-best.

Method	MIMIC-III Mortality			MIMIC-IV Readmission		
	AUROC↑	AUPRC↑	F1↑	AUROC↑	AUPRC↑	F1↑
SMART (fine-tuned predictor)	80.64	45.40	47.18	74.41	54.62	57.38
GraphWalker (with SMART encoder)	**84.19**	**59.46**	**58.24**	**75.91**	**60.61**	**58.01**
Δ	+3.55	+14.06	+11.06	+1.50	+5.99	+0.63
ConCare (fine-tuned predictor)	69.19	36.01	40.65	70.30	41.87	50.26
GraphWalker (with ConCare encoder)	**70.46**	**36.93**	**45.90**	**77.01**	**59.49**	**59.44**
Δ	+1.27	+0.92	+5.25	+6.71	+17.62	+9.18
AdaCare (fine-tuned predictor)	60.67	19.37	26.81	78.40	59.37	54.87
GraphWalker (with AdaCare encoder)	**65.61**	**27.13**	**37.89**	**80.45**	**62.72**	**60.04**
Δ	+4.94	+7.76	+11.08	+2.05	+3.35	+5.17

Table 4: Performance of *GraphWalker* with different EHR encoders on MIMIC-III mortality and MIMIC-IV readmission, using *Qwen3-14B* as the backbone LLM. For each encoder, we compare two settings: (i) the encoder fine-tuned end-to-end as a task-specific supervised predictor, and (ii) *GraphWalker* using the same pretrained encoder (without fine-tuning) for graph-based demonstration selection. Δ denotes the absolute improvement (in percentage points) of *GraphWalker* over the supervised fine-tuned baseline. Note that this differs from *EHR-emb* in Table 1, which uses SMART embeddings for ICL retrieval without fine-tuning.

and dataset-dependent trends, indicating that **simply adding more demonstrations does not necessarily improve EHR reasoning performance**—irrelevant or misleading demonstrations can disrupt rather than enhance coherent reasoning.

***GraphWalker* is the only method whose performance improves consistently with k.** By grounding demonstration selection in cohort-level clinical similarity and applying LLM-guided composition-aware search, *GraphWalker* progressively identifies demonstration sets that maximize information gain rather than adding redundant or harmful cases.

Purely model-perspective methods struggle relative to encoder-based baselines. Methods such as CONE, LMS3, and SPELL consistently underperform *EHR-emb* and *GraphWalker*, suggesting that **LLM-internal signals alone are insufficient to construct reliable demonstration sets for EHR reasoning** without grounding in clinically meaningful patient distributions.

C.2.4 Runtime Analysis

The central design objective of *GraphWalker* is to improve clinical reasoning accuracy by deliberately allocating a small additional inference budget at test time. In clinical decision-support scenarios, predictive accuracy often carries far greater practical value than marginal reductions in latency, since higher-quality predictions can directly affect the quality of downstream clinical decisions. Motivated by this, *GraphWalker* invests extra inference cost to construct demonstration sets that are clinically coherent, complementary, and informative for each target patient—following the same high-level principle as test-time scaling in reasoning models, where additional computation is intentionally ex-

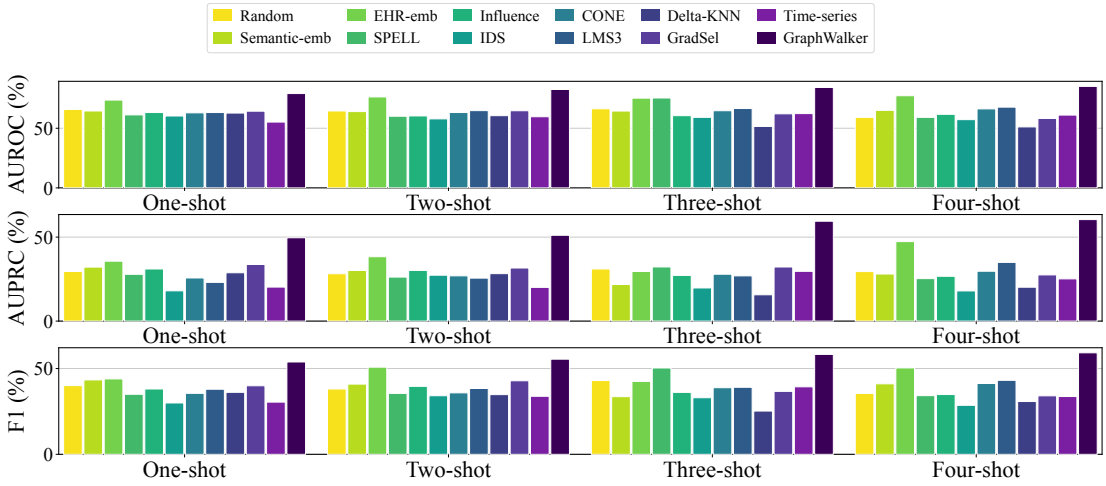

Figure 8: Shot scaling performance on MIMIC-III Mortality with Qwen3-14B as the backbone LLM.

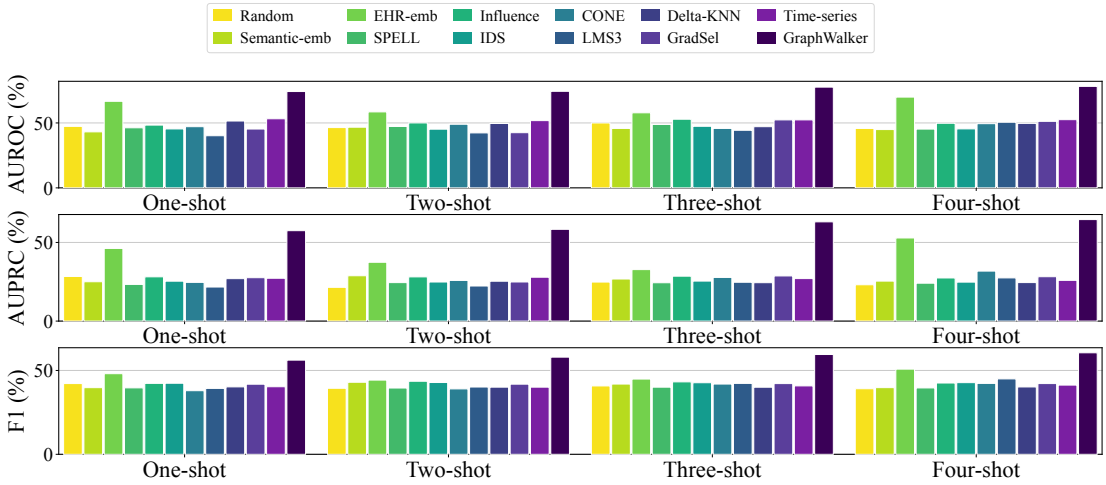

Figure 9: Shot scaling performance on MIMIC-IV Readmission with *Qwen3-14B* as the backbone LLM.

changed for better reasoning outcomes.

Crucially, this extra cost is tightly bounded rather than spent on unconstrained global search. Cohort retrieval, anchor initialization, frontier expansion, lazy greedy selection, and early stopping jointly restrict the inference budget to a small set of clinically relevant candidates, making *GraphWalker* a targeted form of test-time demonstration optimization.

Experiments. We conduct runtime analysis on *MIMIC-III Mortality* using *Qwen3-14B* as the backbone LLM. For each demonstration selection method, we measure the average wall-clock evaluation time per test case—including both demonstration selection and final prediction generation—and compare the resulting performance–latency trade-off across methods.

Results. As shown in Figure 10, *GraphWalker* achieves the strongest performance across all three metrics while keeping inference cost on par with existing LLM-based selection methods. The search overhead is effectively bounded by the cohort-guided frontier design rather than expanded into exhaustive global exploration. Two observations make this concrete:

(i) The absolute overhead is modest. Compared with the strongest similarity-based baseline *EHR-emb*, *GraphWalker* incurs only 7.2s additional inference time per case (13.9s vs. 6.7s), yet yields substantial gains of +9.0 AUROC, +29.9 AUPRC, and +15.8 F1.

(ii) The overhead is competitive among LLM-based selection methods. Among approaches that similarly invoke LLM-side scoring, *GraphWalker* (13.9s) is on par with *CONE* (13.1s), and notably faster than *Influence* (14.5s), *IDS* (16.3s), and

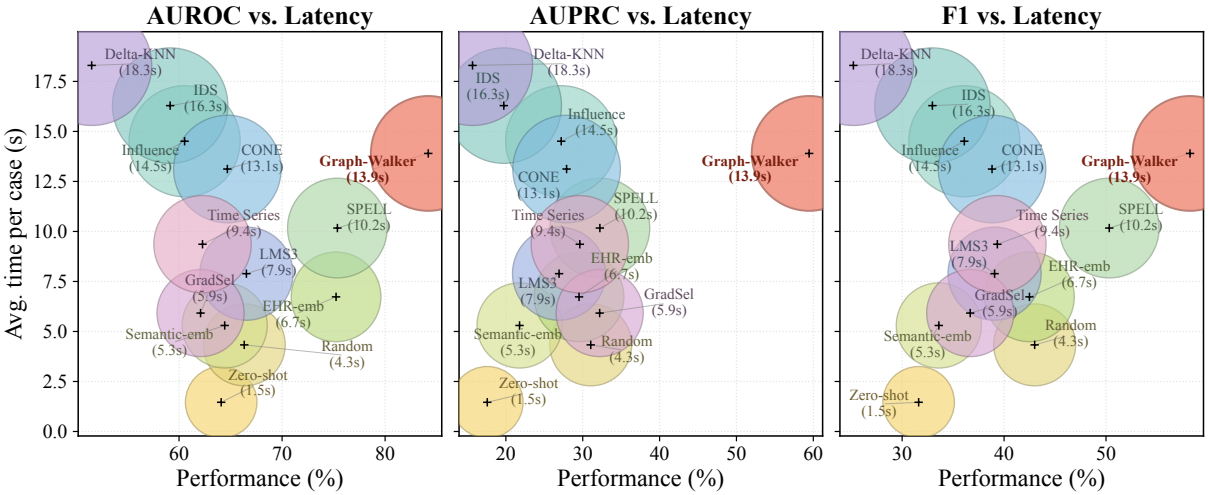

Figure 10: Performance–latency tradeoff on MIMIC-III mortality across AUROC, AUPRC, and F1 (left to right). Each bubble represents one case retrieval method, with position showing mean performance (x-axis) and average inference time per case (y-axis); bubble color identifies the method. *GraphWalker* (red) achieves the strongest performance across all three metrics while keeping latency comparable to prior LLM-based case selection methods.

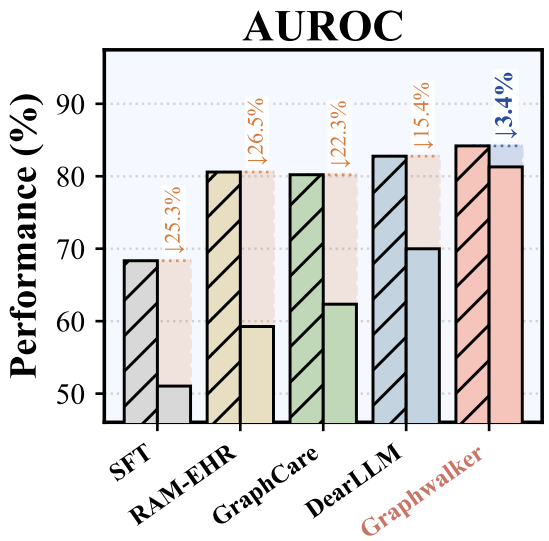

Figure 11: In-domain vs. cross-dataset OOD AUROC on MIMIC-III in-hospital mortality, complementing Figure 7 in the main text. Hatched bars show in-domain performance (trained and tested on MIMIC-III); solid bars show OOD performance (trained on MIMIC-IV, tested on MIMIC-III without fine-tuning); shaded regions show the relative drop. *GraphWalker* shows a smaller drop than the supervised baselines we evaluated.

Delta-KNN (18.3s), while delivering substantially larger accuracy gains across all three metrics.

Together, these results show that ***GraphWalker* converts a limited test-time cost into significant improvements in clinical prediction quality.**

C.3 RQ3 Results

We evaluate all methods on **in-hospital mortality prediction** on MIMIC-III in two regimes:

In-Domain. Each supervised baseline is trained on the MIMIC-III training set and evaluated on the MIMIC-III test set, following the standard supervised protocol. For *GraphWalker*, the patient graph and cohort partitions are constructed on the MIMIC-III training set, and predictions are made on the MIMIC-III test set without any parameter updates.

Cross-Dataset OOD. Each supervised baseline is trained on the MIMIC-IV training set and evaluated on the MIMIC-III test set *without any fine-tuning*. For *GraphWalker*, the patient graph, cohort partitions, and case retrieval base are all constructed on the MIMIC-IV training set, while predictions remain on the MIMIC-III test set. The EHR encoder SMART is a **frozen pretrained model** that is not fine-tuned for any specific dataset, enabling cross-dataset patient embedding comparison without retraining; the frozen LLM backbone and all other *GraphWalker* components likewise remain unchanged across both regimes.

Baselines. We compare against four supervised paradigms:

- **SFT** fine-tunes the LLM backbone on the source training set with next-token prediction loss to generate the binary mortality label.
- **RAM-EHR** (Xu et al., 2024a) enhances supervised prediction by retrieving knowledge from multiple sources and applying consistency regularization.
- **GraphCare** (Jiang et al., 2024) acquires personalized knowledge by directing LLMs to produce triples, which are then fed into a graph neural network predictor.
- **DearLLM** (Xu et al., 2025) captures quantitative

Method	AUROC			AUPRC			F1		
	In-Dom.	OOD	Drop	In-Dom.	OOD	Drop	In-Dom.	OOD	Drop
SFT	68.34±6.29	51.04±5.47	↓25.3%	17.48±4.88	11.34±4.11	↓35.1%	30.43±6.35	17.38±5.12	↓42.9%
RAM-EHR (Xu et al., 2024a)	80.58±3.48	59.26±3.35	↓26.5%	53.02±3.87	39.23±4.47	↓26.0%	51.26±3.44	37.26±3.67	↓27.3%
GraphCare (Jiang et al., 2024)	80.21±3.55	62.34±5.57	↓22.3%	54.77±3.19	39.35±3.67	↓28.2%	52.22±3.75	36.04±4.36	↓31.0%
DearLLM (Xu et al., 2025)	82.77±3.63	69.99±5.21	↓15.4%	58.02±3.24	42.85±6.39	↓26.1%	56.79±3.75	43.32±6.06	↓23.7%
GraphWalker	**84.19±3.35**	**81.29±3.91**	**↓3.4%**	**59.46±7.64**	**52.98±6.01**	**↓10.9%**	**58.24±5.89**	**54.08±5.84**	**↓7.1%**

Table 5: Performance comparison of *GraphWalker* against supervised methods on MIMIC-III in-hospital mortality. "In-Dom." refers to training and testing on MIMIC-III; "OOD" refers to training on MIMIC-IV and testing on MIMIC-III without fine-tuning. "Drop" shows the relative degradation from In-Domain to OOD. *GraphWalker* requires no parameter updates in either regime.

feature correlations through conditional perplexity of LLM deduction, augmenting supervised predictors via a feature-frequency-aware graph pooling method.

All baselines and *GraphWalker* use *Qwen3-14B* as the LLM backbone for fair comparison; implementations follow the original papers.

Results. Table 5 reports detailed in-domain and cross-dataset OOD performance, complementing Figure 7 in the main text. We summarize three observations:

(i) In-domain, GraphWalker outperforms all four supervised baselines on every metric. For example, against the strongest supervised baseline DearLLM, *GraphWalker* attains 84.19 vs. 82.77 AUROC and 59.46 vs. 58.02 AUPRC, despite requiring no parameter updates.

(ii) Under distribution shift, all supervised methods show notable degradation. The drop ranges from ↓15.4% to ↓42.9% across metrics, with SFT degrading the most and DearLLM the least. Notably, even hybrid pipelines that leverage LLM-derived knowledge exhibit a comparable level of degradation: in these methods, the LLM is used as an upstream knowledge source while a downstream supervised predictor handles the final prediction, so the overall pipeline inherits the generalization limits of its supervised component. The reliance on fixed input schemas (feature order, availability, and encoding) further renders these models brittle to schema variations across institutions with heterogeneous EHR systems.

(iii) GraphWalker shows substantially smaller degradation, suggesting it learns more transferable clinical patterns than feature-level correlations. The drop ranges from ↓3.4% (AUROC) to ↓10.9% (AUPRC), noticeably smaller than that of the supervised baselines we evaluated. Beyond the absence of source-specific parameter updates, this robustness reflects a more fundamental property of our approach: rather than fitting shallow feature-level correlations specific to the source dataset, *GraphWalker* performs *reasoning by analogy over retrieved patient cases*, asking the LLM to interpret each test patient by analogy to clinically similar past patients. This case-level reasoning relies on transferable clinical patterns—such as disease progression and comorbidity dynamics—that generalize more naturally across hospitals, populations, and EHR schemas than feature-level statistical correlations.

C.4 RQ4 Results

We investigate two practical questions beyond the main experiments: (i) **whether *GraphWalker* can be extended to closed-source or black-box LLMs** where token-level probabilities are not accessible, and (ii) **whether *GraphWalker* can be combined with agentic reasoning frameworks** such as ReAct to yield further gains under the same few-shot setting. We implement a black-box variant of *GraphWalker* (Sec C.4.1) and integrate it with a ReAct-style scaffold (Sec C.4.2).

C.4.1 Black-box GraphWalker

The original *GraphWalker* estimates the marginal information gain of a candidate demonstration via the change in conditional entropy computed from model probabilities. While effective for open-source LLMs with accessible logits, this design is not directly applicable to closed-source frontier models exposed only through API-based text generation. To address this, we develop a black-box variant that replaces entropy-based scoring with LLM self-evaluation.

Pipeline. The graph construction and search structure remain unchanged. We build a patient KNN graph from the training set using pretrained EHR embeddings, then apply Leiden clustering to obtain patient cohorts. For each test patient, we retrieve the top-L most relevant cohorts and initialize anchor nodes by selecting the top-K most simi-

Method	Direct Answer (Few-shot)			+ ReAct		
	AUROC	AUPRC	F1	AUROC	AUPRC	F1
Zero-shot	70.81±4.86	33.69±6.88	41.42±6.20	71.11±5.47	32.86±4.97	43.61±6.33
Random	70.41±4.69	31.32±6.94	39.12±5.61	68.46±3.26	29.08±5.53	39.08±4.96
Semantic-emb	69.91±5.23	25.78±6.32	36.30±6.09	70.91±4.11	26.01±5.37	36.92±5.58
EHR-emb	81.59±4.87	48.45±6.96	46.07±5.78	82.09±7.81	47.98±5.96	48.51±6.45
GraphWalker	**87.64**±5.14	**62.31**±8.96	**64.79**±7.74	**88.72**±3.18	**64.69**±2.43	**67.11**±2.45

Table 6: Results on MIMIC-III Mortality with GPT-5.2 as the black-box LLM, comparing two inference settings under the *same* few-shot demonstrations: **Direct Answer (Few-shot)** where the LLM directly outputs the prediction with no reasoning scaffold, and **+ ReAct** where the LLM applies ReAct-style intermediate reasoning before generating the prediction. Bold = best within column, underline = second-best. Higher is better (↑).

lar patients within each cohort, forming the initial frontier for graph-guided demonstration search.

The key difference lies in the greedy selection step. Instead of computing ΔH from token-level probabilities, we score each frontier candidate by querying the black-box LLM with a self-evaluation prompt. Given the target patient, the already selected demonstrations, and one candidate demonstration, the model rates how much *additional useful information* the candidate provides for predicting the target patient's outcome on an integer scale from 0 to 10.

Greedy selection with self-evaluation. At each iteration, all frontier candidates are scored in parallel through API calls. We select the highest-scoring candidate, add it to the demonstration set, and expand the frontier with its graph neighbors. The process repeats until either the maximum number of demonstrations is reached or the highest candidate score becomes zero. This preserves the cohort-guided frontier expansion and greedy search structure of *GraphWalker*, while replacing white-box entropy computation with a black-box scoring proxy.

Scoring prompt. The scoring prompt (Table C.4.1) approximates the same intuition as the original information-gain objective: a useful candidate should be clinically relevant to the test patient, provide complementary evidence beyond the already selected examples, and reduce uncertainty about the target prediction. Concretely, the black-box scorer assesses each candidate from three aspects: (1) clinical similarity or complementarity to the target patient, (2) whether it contributes information not already covered by selected examples, and (3) how much it helps reduce predictive uncertainty. The model outputs only a single integer score from 0 to 10.

Black-box GraphWalker Scoring Prompt

You are evaluating whether a candidate patient example would be helpful for predicting a test patient's clinical outcome.
Task:
`{TASK_DESCRIPTION}`
Already Selected Examples for In-Context Learning:
`{SELECTED_EXAMPLES}`
Candidate Example to Evaluate:
`{CANDIDATE_EXAMPLE}`
Test Patient to Predict:
`{TEST_PATIENT_DETAIL}`
Instruction:
Rate how much *additional useful information* the candidate example provides for predicting the test patient's outcome, considering:

- Clinical similarity or complementarity to the test patient
- Whether it provides new information not already covered by selected examples
- How much it would help reduce uncertainty about the test patient's outcome

Score Definition (0–10):

- `0`: completely redundant or irrelevant
- `5`: moderately helpful, provides some new perspective
- `10`: extremely informative, highly relevant and complementary

Output Format:
Output **ONLY** a single integer score from `0` to `10`, and nothing else.

Discussion. This black-box implementation does not claim to exactly reproduce the white-box entropy objective. Instead, it provides a practical approximation that preserves the core design principle of *GraphWalker*: selecting demonstrations based on *target-conditioned marginal utility* rather than static similarity alone. As a result, *GraphWalker* extends to stronger closed-source LLMs without requiring access to internal logits, broadening its applicability in realistic deployment settings.

Experiments. We evaluate the black-box variant under a standard few-shot setting on *MIMIC-III Mortality*, where the downstream LLM directly outputs the final prediction without any additional reasoning scaffold.

Results. As shown in Table 6, we summarize three observations:

(i) Generic semantic similarity is actively harmful under black-box settings. *Semantic-emb* not only substantially underperforms the zero-shot baseline on AUPRC and F1 (e.g., AUPRC 25.78 vs. 33.69), but is also worse than *Random* retrieval (25.78 vs. 31.32 on AUPRC). This indicates that surface-level semantic similarity, when divorced from clinical structure, can mislead demonstration selection rather than help it.

(ii) Clinically informed similarity helps but remains single-perspective. *EHR-emb* provides a stronger retrieval signal by capturing clinically meaningful patient similarity, clearly outperforming both zero-shot and random retrieval. Yet as a purely similarity-based method, it still leaves substantial headroom relative to *GraphWalker*.

(iii) Black-box GraphWalker substantially outperforms all single-perspective baselines. Compared with the strongest baseline *EHR-emb*, black-box *GraphWalker* improves AUPRC from 48.45 to 62.31. This confirms that its dual-view advantage, which bridges cohort-level clinical similarity and model-side utility estimation, generalizes beyond the original entropy-based implementation: the core strength lies in cohort-guided, target-conditioned demonstration construction rather than any specific scoring interface.

C.4.2 Compatibility with Agentic Reasoning

Recent LLM agents show strong reasoning and decision-making across many domains (Ferrag et al., 2025). Rather than positioning *GraphWalker* as an alternative to agentic reasoning, we view it as a **complementary upstream plugin**: even powerful agent-style reasoning depends on the quality of the provided case context, and *GraphWalker*'s cohort-guided, target-conditioned demonstration selection can supply more informative examples to better unlock downstream agentic reasoning.

Experiments. We choose *ReAct* (Yao et al., 2022) as a representative agentic reasoning framework: it tightly couples intermediate reasoning with final decision making and is a widely used backbone for dynamic agent-style inference. We keep the few-shot demonstrations selected by each method unchanged and only replace the downstream inference style from direct answer generation to ReAct-style reasoning. This design isolates the effect of combining *GraphWalker* with agentic reasoning, directly testing whether better case selection can further enhance a more dynamic reasoning process under the same few-shot setting. Experiments are conducted on *MIMIC-III Mortality* using the same black-box LLM and demonstration selection baselines as in the previous subsection.

Results. As shown in Table 6, we summarize three observations:

(i) GraphWalker + ReAct further improves over standalone GraphWalker across all metrics. Compared with the standard black-box setting, *GraphWalker* + ReAct improves AUROC from 87.64 to 88.72, AUPRC from 62.31 to 64.69, and F1 from 64.79 to 67.11.

(ii) ReAct does not consistently help, and sometimes hurts, weaker baselines. For *Random*, ReAct *actively degrades* performance across all three metrics (e.g., AUROC 70.41 $\rightarrow$ 68.46, AUPRC 31.32 $\rightarrow$ 29.08). For *Semantic-emb*, changes are marginal. For the stronger *EHR-emb*, gains are mixed: AUROC and F1 improve slightly while AUPRC drops (48.45 $\rightarrow$ 47.98).

(iii) Demonstration quality is a prerequisite for agentic reasoning to be effective. Agentic reasoning compounds positively with high-quality demonstrations (consistent improvements for *GraphWalker*) but compounds neutrally or even negatively with low-quality ones. By constructing more informative and clinically coherent demonstrations, *GraphWalker* provides a stronger evidence base on which agentic reasoning can build, positioning it not as an alternative to agentic reasoning but as a pluggable upstream module.

C.5 Applicability to More Tasks

To further examine the generality of *GraphWalker* beyond EHR-centric prediction tasks, we extend our evaluation to three additional medical reasoning benchmarks: **MedQA** (Jin et al., 2021), **CMB** (Wang et al., 2024), and **CMB-Clin** (Wang et al., 2024).

MedQA consists of medical examination questions designed to assess diagnostic reasoning and clinical decision-making, with each question grounded in a realistic patient case description. Although the full benchmark is multilingual, we fo-

cus on the English subset, which contains approximately 12.7K case-based questions and has been widely adopted for evaluating LLM-based medical reasoning.

CMB is a large-scale Chinese medical benchmark covering a broad range of clinical specialties and question types. Following (Ding et al., 2025b), we filter the dataset to retain only case-based questions involving explicit patient scenarios, ensuring its suitability for in-context reasoning and demonstration-based evaluation.

CMB-Clin (Wang et al., 2024) is the inaugural multi-round medical question-answering benchmark constructed from real-world, complex diagnosis and treatment records. Compared with single-turn medical QA benchmarks, CMB-Clin presents more realistic and longitudinal clinical reasoning challenges, requiring models to handle multi-round interactions, evolving clinical conditions, and more comprehensive diagnostic and treatment decision processes.

Dataset statistics are summarized in Table 7.

Split	MedQA	CMB	CMB-Clin
Train	10,178	15,465	83
Val	1,272	1,940	42
Test	1,273	1,935	83
Total	12,723	19,340	208

Table 7: Dataset statistics for additional medical reasoning benchmarks: MedQA, CMB, and CMB-Clin.

C.5.1 Experimental Setup

In this experimental setting, both *GraphWalker* and the Semantic-embedding baseline use *Qwen3-Embedding-8B* as the retrieval encoder. For fair comparison, all methods share the same backbone LLM, namely *Qwen3-14B*.

C.5.2 Results Analysis

Overall, the results in Table 8 demonstrate that *GraphWalker* generalizes well beyond EHR-centric prediction settings and remains consistently effective across diverse medical reasoning tasks. On the two multiple-choice benchmarks, *GraphWalker* achieves the best EM on both CMB (84.05%) and MedQA (61.10%), outperforming all compared baselines, including strong retrieval-based and influence-based selection methods. This indicates that ***GraphWalker* is not limited to structured EHR classification, but can also transfer effectively to broader medical reasoning scenarios requiring diagnostic understanding and clinical decision-making.**

More importantly, *GraphWalker* also shows strong generalization to the open-ended generation benchmark CMB-clin, where it achieves the best performance on all three generation metrics, including BLEU-1, BLEU-4, and ROUGE-L. The gains on CMB-clin suggest that **the benefits of *GraphWalker* are not confined to answer selection or label prediction, but also extend to multi-round, free-form medical response generation**, where models must produce more comprehensive and coherent answers grounded in complex clinical contexts.

Taken together, these results verify that ***GraphWalker* is a generally applicable in-context demonstration selection framework for medical reasoning, rather than a task-specific solution tailored only to EHR analysis**. Its consistent improvements across both discriminative and generative settings highlight its robustness and suggest that the underlying cohort-guided retrieval and example composition mechanism captures transferable reasoning patterns that remain beneficial across different task formats and evaluation protocols.

D Case Study

To qualitatively illustrate why *GraphWalker* yields more reliable in-context demonstrations for EHR prediction, we present in Table 9 a representative case from the *MIMIC-III Mortality* test set using *Qwen3-14B* as the backbone LLM, where the *Zero-shot* baseline produces a severely miscalibrated prediction, while *GraphWalker* corrects it.

E Experimental Setup of Pilot Study

To investigate the behavior of different ICL strategies for EHR reasoning, we adopt *Qwen3-14B* (Yang et al., 2025) as the backbone LLM for all pilot experiments. We conduct experiments on two widely used, large-scale public EHR benchmarks: *MIMIC-III* (Johnson et al., 2016) and *MIMIC-IV* (Johnson et al., 2023), and focus on two representative downstream tasks: *mortality prediction* and *readmission prediction*. For each task, the LLM is prompted in a few-shot setting using demonstrations selected by different ICL strategies. Detailed dataset statistics, preprocessing procedures, task definitions, and prompt templates are provided in Appendix B.1 and H.

Method	CMB EM	MedQA EM	CMB-clin BLEU-4	CMB-clin BLEU-1	CMB-clin ROUGE-L
Zero-shot	80.21±0.92	56.56±3.59	21.06±1.37	59.85±2.30	44.97±1.73
Random	81.50±2.78	52.96±3.53	26.85±1.44	66.05±2.08	38.63±1.77
Semantic-emb	80.52±2.70	59.58±3.52	26.86±1.47	64.90±2.01	41.26±2.07
Time-series	78.05±2.90	52.11±3.55	29.05±10.65	66.56±2.24	40.12±1.57
SPELL	81.98±2.73	54.10±3.49	28.41±1.46	66.75±2.10	36.82±1.64
Influence	83.50±2.64	53.02±3.50	23.88±1.18	67.81±1.08	47.05±1.57
IDS	79.57±2.81	52.59±3.57	22.63±1.53	43.79±3.08	25.58±1.19
CONE	80.41±2.82	59.17±3.51	29.04±1.49	68.10±1.86	41.10±1.51
LMS3	82.02±2.78	54.09±3.60	26.78±1.51	62.58±2.51	38.02±1.72
Delta-KNN	83.02±2.66	53.47±3.42	23.09±1.37	63.73±1.84	46.50±1.79
GradSel	81.48±2.75	52.93±3.47	24.83±1.45	64.98±2.08	45.13±1.78
Ours	**84.05**±2.58	**61.10**±3.46	**29.68**±1.46	**71.51**±1.26	**47.48**±1.38

Table 8: Performance comparison on CMB, MedQA, and CMB-clin using *Qwen3-14B* as the backbone LLM. CMB and MedQA are evaluated by exact match (EM), while CMB-clin is evaluated by BLEU-4, BLEU-1, and ROUGE-L. Higher values indicate better performance (↑). We highlight the **best** and second best results.

Definition of Information Gain ΔH. To quantitatively analyze how different demonstration selection strategies affect the model's understanding of a test query, we measure information gain using an entropy-based criterion. Unlike the marginal gain formulation used in our method, the ΔH reported in the pilot study is defined *relative to the zero-shot baseline*.

Formally, let x_{test} denote an unlabeled test input, and let $\mathcal{S}_k$ denote a demonstration set of size k selected by a given ICL strategy. We define the conditional entropy of the model's prediction on x_{test} under a demonstration composition $\mathcal{S}_k$ as

$$H_\theta(x_{\text{test}} \mid \mathcal{S}_k) = H_\theta(x_{\text{test}}, \mathcal{S}_k) - H_\theta(\mathcal{S}_k), \quad (7)$$

where $H_\theta(x_{\text{test}}, \mathcal{S}_k)$ denotes the cross-entropy of the full prompt (including both the demonstrations and the test query), and $H_\theta(\mathcal{S}_k)$ denotes the cross-entropy of the demonstration composition alone. This conditional entropy reflects the model's predictive uncertainty and thus its degree of understanding of the test query under the given context.

We then define the information gain of a demonstration composition $\mathcal{S}_k$ relative to the zero-shot setting as

$$\Delta H_\theta(\mathcal{S}_k) = H_\theta(x_{\text{test}} \mid \emptyset) - H_\theta(x_{\text{test}} \mid \mathcal{S}_k), \quad (8)$$

where $\emptyset$ denotes the empty demonstration set (i.e., zero-shot prompting). A larger $\Delta H_\theta(\mathcal{S}_k)$ indicates that the selected demonstrations reduce the model's uncertainty more effectively compared to the zero-shot baseline.

For each ICL strategy and each $k \in \{1, 2, 3, 4\}$, we compute $\Delta H_\theta(\mathcal{S}_k)$ for every validation instance and report the average information gain over the validation set.

F Algorithm

Please see Algorithm 1. For clarity, Algorithm 1 presents the full greedy implementation of *Graph-Walker*. In practice, we adopt a lazy greedy strategy to improve computational efficiency, which is discussed in detail in Appendix G.

G Lazy Greedy Search: Theoretical Justification and Implementation

Lazy greedy is provably exact when the set objective is monotone submodular (Minoux, 2005). However, monotone submodularity over the entire power set is a strong condition that is difficult to verify in closed form for LLM-based objectives. Below we identify a strictly weaker condition under which lazy greedy remains exact in our setting, discuss approximation guarantees beyond this condition, and provide complexity analysis and implementation details.

Sufficient Condition: Chain-wise Diminishing Returns. The exactness of lazy greedy as an *algorithm* does not require submodularity over the entire power set; it only uses cached marginal gains along the specific chain of selected sets produced by the search. We formalize this observation as follows:

Assumption A (Chain-wise Diminishing Returns). For any candidate $v \in \mathcal{V}$ and any chain $\mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \cdots \subseteq \mathcal{S}_K$ of subsets produced by greedy selection, the marginal information gain is non-increasing along the chain:

$$\Delta_q(v \mid \mathcal{S}_t) \geq \Delta_q(v \mid \mathcal{S}_{t+1}), \quad \forall t = 0, 1, \ldots, K-1,\ v \notin \mathcal{S}_{t+1}.$$

Proposition 1 (Lazy Greedy Exactness). *Under Assumption A, lazy greedy returns the same case set as full greedy.*

Proof. We prove by induction on the iteration t that the case sets produced by lazy greedy and full greedy coincide, i.e., $\mathcal{S}_t^{\text{lazy}} = \mathcal{S}_t^{\text{full}}$ for all $t = 0, 1, \ldots, K$.

Base ($t = 0$). Both algorithms start from $\mathcal{S}_0 = \emptyset$.

Inductive step. Assume $\mathcal{S}_{t-1}^{\text{lazy}} = \mathcal{S}_{t-1}^{\text{full}}$, and denote their common value as $\mathcal{S}_{t-1}$. At iteration t, full greedy selects

$$v_t^{\text{full}} = \arg\max_{v \in F_{t-1}} \Delta_q(v \mid \mathcal{S}_{t-1}),$$

where F_{t-1} is the current frontier and the marginal gain is computed exactly for every candidate.

For lazy greedy, consider any candidate v in the priority queue whose cached gain was computed at some earlier iteration $t' < t$, i.e., $\Delta_q^{(\text{cache})}(v) = \Delta_q(v \mid \mathcal{S}_{t'})$. By the inductive hypothesis, $\mathcal{S}_{t'}^{\text{lazy}} = \mathcal{S}_{t'}^{\text{full}} \subseteq \mathcal{S}_{t-1}$. Applying Assumption A iteratively along the chain $\mathcal{S}_{t'} \subseteq \mathcal{S}_{t'+1} \subseteq \cdots \subseteq \mathcal{S}_{t-1}$ yields

$$\Delta_q^{(\text{cache})}(v) = \Delta_q(v \mid \mathcal{S}_{t'}) \geq \Delta_q(v \mid \mathcal{S}_{t-1}),$$

so every cached gain in the priority queue is a valid upper bound on the current marginal gain.

Let v_{top} denote the candidate with the largest cached gain. Lazy greedy recomputes its exact gain $g_{\text{top}}^{\text{exact}} = \Delta_q(v_{\text{top}} \mid \mathcal{S}_{t-1})$ and accepts it as v_t^{lazy} when $g_{\text{top}}^{\text{exact}}$ is no smaller than the next cached value $g_{\text{next}}^{\text{cache}}$ in the queue. By the upper bound property, for every other candidate $v' \in F_{t-1} \setminus \{v_{\text{top}}\}$,

$$g_{\text{next}}^{\text{cache}} \geq \Delta_q^{(\text{cache})}(v') \geq \Delta_q(v' \mid \mathcal{S}_{t-1}).$$

Combining these inequalities,

$$g_{\text{top}}^{\text{exact}} \geq g_{\text{next}}^{\text{cache}} \geq \Delta_q(v' \mid \mathcal{S}_{t-1}), \quad \forall v' \in F_{t-1} \setminus \{v_{\text{top}}\},$$

so v_{top} achieves the maximum exact marginal gain over F_{t-1}. Therefore $v_t^{\text{lazy}} = v_t^{\text{full}}$, and $\mathcal{S}_t^{\text{lazy}} = \mathcal{S}_t^{\text{full}}$. □

Assumption A is strictly weaker than monotone submodularity over the entire power set: it requires diminishing returns only along the chain of subsets actually encountered by the algorithm, not for all subset pairs. Our analysis in Section 2 (Figure 4) is consistent with Assumption A: marginal gains exhibit clear diminishing returns as more cases are added, with information-gain-based methods saturating monotonically and semantic-similarity-based methods decaying due to redundancy among clinically similar cases.

Approximation Guarantee Beyond Assumption A. For settings where Assumption A may be violated, the broader theory of approximate submodular maximization provides formal guarantees. The submodularity ratio of f_q is defined as

$$\gamma_q = \min_{\mathcal{S} \subseteq \mathcal{V},\, v \notin \mathcal{S}} \frac{\Delta_q(v \mid \mathcal{S})}{\Delta_q(v \mid \emptyset)},$$

where $\gamma_q = 1$ recovers strict submodularity. Greedy selection achieves a $(1 - e^{-\gamma_q})$ approximation of the optimal case set, which degrades gracefully as γ_q deviates from 1. This provides a formal fallback guarantee for the approximate-submodular regime, ensuring that the quality of the lazy greedy solution remains bounded even when Assumption A holds only approximately.

Motivation and Complexity. A naive implementation of Algorithm 1 performs a *full greedy* search, recomputing the marginal information gain $\Delta_q(\cdot \mid \mathcal{S}_{t-1}^{(q)})$ for every candidate in the frontier at each iteration. Let $|F_t^{(q)}|$ denote the frontier size at step t and K be the case budget. Full greedy therefore requires $\sum_{t=1}^{K} |F_t^{(q)}|$ exact evaluations, which translates to $O(K\bar{F})$ LLM forward passes, where $\bar{F}$ denotes the average frontier size. Since each exact evaluation requires a forward pass over the in-context prompt, this cost becomes prohibitive for large patient graphs.

Under Proposition 1 (or approximate submodularity), lazy greedy reduces this to approximately $O(|F_0^{(q)}| + R)$ exact evaluations, where R is the number of cached entries that require re-evaluation during the search. The priority-queue overhead is $O(\log |F_t^{(q)}|)$ per update, which is negligible compared to LLM inference.

Implementation Details. Our implementation follows the standard lazy greedy template with a max-priority queue. At each step, we pop the candidate with the largest *cached* marginal gain (an

upper bound under Proposition 1), recompute its *exact* marginal gain with respect to the current case set $\mathcal{S}^{(q)}$, and compare it with the next cached best. If it remains the best, we accept it; otherwise, we update its cached key and push it back. Upon selecting v^*, we update $\mathcal{S}^{(q)} \leftarrow \mathcal{S}^{(q)} \cup \{v^*\}$, remove v^* from the frontier, and expand the frontier by adding $\mathcal{N}(v^*) \backslash \mathcal{S}^{(q)}$. We further apply early stopping when the best marginal gain becomes non-positive, i.e., $\Delta_q(v^* \mid \mathcal{S}^{(q)}) \leq 0$, preventing the inclusion of misleading or uninformative cases.

H Prompt Details

In this section, we detail the prompt designs used in our framework. We implement two prompting schemes: a *zero-shot* template, which provides no in-context examples and requires the LLM to directly produce the task-specific output, and a *few-shot* template, which augments the prompt with a set of labeled patient demonstrations to support ICL. Both templates share a unified structure for presenting longitudinal EHR data. In addition, we specify task-specific instructions for each prediction task, including in-hospital mortality, readmission, and length-of-stay prediction, ensuring consistent and fair evaluation across different prompting settings.

Prompt for Mortality Prediction

You are tasked with predicting in-hospital mortality based on patient EHR data.
Provide only a floating-point number between 0 and 1 representing the predicted probability of mortality (a higher value indicates a higher likelihood of death).
Do **not** provide any reasoning, explanation, or additional text. Output **only** the numerical value.
Example: `0.XX`

Prompt for Readmission Prediction

You are tasked with predicting whether a patient will be readmitted within 30 days after hospital discharge based on EHR data.
Provide only a floating-point number between 0 and 1 representing the predicted probability of 30-day readmission after discharge (including cases where the patient dies within 30 days, which are counted as readmission events).
Do **not** provide any reasoning, explanation, or additional text. Output **only** the numerical value.
Example: `0.XX`

Prompt for Length-of-Stay (LOS) Prediction

You are tasked with predicting the patient's length of hospital stay based on EHR data.
Provide only a single letter (**A**, **B**, **C**, or **D**) representing the predicted length-of-stay category:

- **A**: Less than 3 days ($<$ 3 days)
- **B**: 3 to 7 days (3–7 days)
- **C**: 7 to 14 days (7–14 days)
- **D**: More than 14 days ($>$ 14 days)

Do **not** provide any reasoning, explanation, or additional text. Output **only** the letter (**A**, **B**, **C**, or **D**).
Example: `B`

Zero-shot Prompt Template

You will be provided with longitudinal electronic health record (EHR) data for a single patient. Each clinical feature is represented as a time-ordered list of measurements corresponding to the same hospital stay. Missing values are denoted as NaN. Units and reference ranges are provided where applicable.
Patient Information:

- Number of measurements: {LENGTH}
- Measurement times (hours since admission): [{RECORD_TIME_LIST}]

Task Description:
{TASK_DESCRIPTION}
Clinical Features Over Time:

- **Heart Rate** (Unit: bpm. Reference range: 60–100):
  [82, 86, 90, 95, NaN, 88, 84, . . .]
- **Systolic Blood Pressure** (Unit: mmHg. Reference range: < 120):
  [110, 118, 125, 132, 120, NaN, 115, . . .]
- **Oxygen Saturation** (Unit: %. Reference range: 95–100):
  [100, 99, 98, NaN, 96, 95, 97, . . .]
- **Glasgow Coma Scale (Total)** (Unit: /. Reference range: /.):
  [15, 14, NaN, 12, 10, 8, . . .]
- ⋮

Your Answer:
{MODEL_OUTPUT}

Few-shot Prompt Template

You will be provided with longitudinal electronic health record (EHR) data for a single patient. Each clinical feature is represented as a time-ordered list of measurements corresponding to the same hospital stay. Missing values are denoted as NaN. Units and reference ranges are provided where applicable.
Patient Information:

- Number of measurements: {LENGTH}
- Measurement times (hours since admission): [{RECORD_TIME_LIST}]

Task Description:
{TASK_DESCRIPTION}
Instructions & Output Format:
{RESPONSE_FORMAT}
In-context Examples:
Below are example patient records and their corresponding labels. These examples are provided to guide the prediction for the target patient.

- **Example 1:**
  Clinical Features: [. . .]
  Label: {LABEL_1}
- **Example 2:**
  Clinical Features: [. . .]
  Label: {LABEL_2}
- ⋮

Target Patient:
Clinical Features Over Time:

- **Heart Rate** (Unit: bpm. Reference range: 60–100):
  [82, 86, 90, 95, NaN, 88, 84, . . .]
- **Systolic Blood Pressure** (Unit: mmHg. Reference range: < 120):
  [110, 118, 125, 132, 120, NaN, 115, . . .]
- **Oxygen Saturation** (Unit: %. Reference range: 95–100):
  [100, 99, 98, NaN, 96, 95, 97, . . .]
- **Glasgow Coma Scale (Total)** (Unit: /. Reference range: /.):
  [15, 14, NaN, 12, 10, 8, . . .]
- ⋮

Your Answer:
{MODEL_OUTPUT}

Algorithm 1 *GraphWalker*

Require: Training EHR dataset $\{\mathbf{X}_i\}_{i=1}^{N}$; test patient record $\mathbf{X}^{(q)}$; pretrained EHR encoder $\mathcal{M}_{\text{exp}}$; LLM M_θ; graph neighbor size k_g; number of retrieved cohorts K_c; anchors per cohort K_a; demonstration budget K.

# Stage 1: Patient Graph Construction and Cohort Discovery
for $i \leftarrow 1$ to N **do**
  $\mathbf{h}_i \leftarrow \mathcal{M}_{\text{exp}}(\mathbf{X}_i)$
end for
Construct a (symmetrized) k_g-NN patient graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over $\{\mathbf{h}_i\}$ using cosine similarity
$\{\mathcal{C}_m\}_{m=1}^{M} \leftarrow \textsc{Leiden}(\mathcal{G})$
for $m \leftarrow 1$ to M **do**
  $\mathbf{z}_m \leftarrow \frac{1}{|\mathcal{C}_m|} \sum_{i \in \mathcal{C}_m} \mathbf{h}_i$
end for
# Stage 2: Instance-aware Cohort Retrieval and Anchor Initialization
$\mathbf{h}^{(q)} \leftarrow \mathcal{M}_{\text{exp}}(\mathbf{X}^{(q)})$
$\mathcal{C}_{\text{ret}}^{(q)} \leftarrow \text{Top-}K_c\big(\text{sim}(\mathbf{h}^{(q)}, \mathbf{z}_m)\big)$
$\mathcal{F}^{(q)} \leftarrow \emptyset$
for each cohort $\mathcal{C}_m \in \mathcal{C}_{\text{ret}}^{(q)}$ **do**
  $\mathcal{V}_m^{(q)} \leftarrow \text{Top-}K_a\big(\text{sim}(\mathbf{h}^{(q)}, \mathbf{h}_i)\big),\ v_i \in \mathcal{C}_m$
  $\mathcal{F}^{(q)} \leftarrow \mathcal{F}^{(q)} \cup \mathcal{V}_m^{(q)}$
end for
# Stage 3: LLM-guided Greedy Search with Frontier Expansion
$\mathcal{S}^{(q)} \leftarrow \emptyset$
$H_0 \leftarrow H_\theta(\mathbf{X}^{(q)} \mid \mathcal{S}^{(q)})$
while $|\mathcal{S}^{(q)}| < K$ **and** $\mathcal{F}^{(q)} \neq \emptyset$ **do**
  $v^* \leftarrow \texttt{None}$; $\Delta^* \leftarrow 0$; $H^* \leftarrow H_0$
  for each $v_i \in \mathcal{F}^{(q)}$ **do**
    $H_i \leftarrow H_\theta(\mathbf{X}^{(q)} \mid \mathcal{S}^{(q)} \cup \{v_i\})$
    $\Delta H_\theta(v_i \mid \mathcal{S}^{(q)}) \leftarrow H_0 - H_i$
    if $\Delta H_\theta(v_i \mid \mathcal{S}^{(q)}) > \Delta^*$ **then**
      $v^* \leftarrow v_i$; $\Delta^* \leftarrow \Delta H_\theta(v_i \mid \mathcal{S}^{(q)})$; $H^* \leftarrow H_i$
    end if
  end for
  if $v^* = \texttt{None}$ **or** $\Delta^* \leq 0$ **then**
    break
  end if
  $\mathcal{S}^{(q)} \leftarrow \mathcal{S}^{(q)} \cup \{v^*\}$
  $H_0 \leftarrow H^*$
  $\mathcal{F}^{(q)} \leftarrow \mathcal{F}^{(q)} \setminus \{v^*\}$
  $\mathcal{F}^{(q)} \leftarrow \mathcal{F}^{(q)} \cup \mathcal{N}(v^*)$
  $\mathcal{F}^{(q)} \leftarrow \mathcal{F}^{(q)} \setminus \mathcal{S}^{(q)}$
end while
return $\mathcal{S}^{(q)}$

Table 9: Case study on *MIMIC-III Mortality*.

Component	Content
Prompt	You will be provided with longitudinal electronic health record (EHR) data for a single patient. Each clinical feature is represented as a time-ordered list of measurements corresponding to the same hospital stay. Missing values are denoted as `NaN`. **Patient Information:** • Number of measurements: `19` • Measurement times (hours since admission): `[4.68, 5.18, 6.18, 7.18, 8.18, . . . ]` **Task Description:** You are tasked with predicting in-hospital mortality based on patient EHR data. **Instructions & Output Format:** Provide only a floating-point number between 0 and 1 representing the predicted probability of mortality. Do **not** provide any reasoning or additional text. Output **only** the numerical value (e.g., `0.XX`).
Target Patient	**Clinical Features Over Time:** • **Heart Rate (bpm):** `[117.0, 110.0, 111.0, 107.0, 102.0, . . . , 119.0, 122.0, 118.0]` • **Systolic BP (mmHg):** `[99.0, 96.0, 92.0, 109.0, 87.0, . . . , 127.0, 118.0, 120.0]` • **Diastolic BP (mmHg):** `[39.0, 56.0, 44.0, 54.0, 47.0, . . . , 87.0, 75.0, 67.0]` • **Mean BP (mmHg):** `[59.0, 69.3, 60.0, 72.3, 60.3, . . . , 100.3, 89.3, 84.7]` • **SpO_2 (%):** `[97.0, 94.0, 95.0, 95.0, 94.0, . . . , 96.0, 95.0, 95.0]` • **GCS (Total):** `[NaN, 15.0, 15.0, 15.0, 15.0, . . . , 15.0, 15.0, 15.0]` ⋮
Ground Truth	`0` (Survival).
Zero-shot Output	**Prediction:** `0.72` → **Incorrect**
***GraphWalker* Examples**	**Example 1** *Label:* `0`*;* • **Heart Rate:** `[78.0, 104.0, 81.0, 98.0, 74.0, . . . , 73.0, NaN, 74.0, 72.0]` • **Systolic BP:** `[119.0, 134.0, 111.0, 165.0, 149.0, . . . , 107.0, 96.0, 112.0, 118.0]` • **Diastolic BP:** `[62.0, 70.0, 79.0, 73.0, 66.0, . . . , 43.0, 56.0, 62.0]` • **SpO_2:** `[98.0, 94.0, 95.0, 98.0, 96.0, . . . , 98.0, NaN, 97.0, 96.0]` • **GCS (Total):** `[NaN, 14.0, NaN, 14.0, NaN, . . . , 15.0, . . . ]` ⋮ **Example 2** *Label:* `0`*;* • **Heart Rate:** `[87.0, 92.0, 92.0, 97.0, 90.0, . . . , 88.0, 77.0, 75.0, 74.0]` • **Systolic BP:** `[138.0, 155.0, 143.0, 152.0, 131.0, . . . , 136.0, 116.0, 133.0, 136.0]` • **Diastolic BP:** `[62.0, 78.0, 69.0, 79.0, 57.0, . . . , 59.0, 66.0, 61.0]` • **SpO_2:** `[93.0, 93.0, 93.0, 95.0, 91.0, . . . , 93.0, 97.0, 98.0, 98.0]` • **GCS (Total):** `[NaN, 15.0, NaN, . . . , 15.0, . . . ]` ⋮ **Example 3** *Label:* `0`*;* • **Heart Rate:** `[102.0, 105.0, 95.0, 91.0, 96.0, . . . , 77.0, 79.0, 80.0, 81.0]` • **Systolic BP:** `[122.0, 118.0, 115.0, 118.0, 106.0, . . . , 132.0, 118.0, 129.0, 130.0]` • **Diastolic BP:** `[62.0, 52.0, 65.0, 74.0, 63.0, . . . , 84.0, 76.0, 79.0, 85.0]` • **SpO_2:** `[83.0, 95.0, 94.0, 95.0, 93.0, . . . , 93.0, 93.0, 92.0, 91.0]` • **GCS (Total):** `[NaN, NaN, 15.0, . . . , 15.0, . . . ]` ⋮
***GraphWalker* Output**	**Prediction:** `0.0689` → **Correct**

I Notations Table

This section summarizes the key notations used in the *GraphWalker* framework.

Symbol	Description
$\mathbf{X}_i$	EHR record of patient i
$\mathcal{M}_{\text{exp}}$	Pretrained EHR encoder
$\mathbf{h}_i$	Patient embedding of patient i
N	Number of patients in the EHR base
$\mathcal{G}$	Population-level patient graph
$\mathcal{V}$	Set of patient nodes in the graph
$\mathcal{E}$	Set of edges constructed via k_g-nearest neighbors in the embedding space
k_g	Number of nearest neighbors used for graph construction
$\text{sim}(\cdot,\cdot)$	Cosine similarity function between embeddings
$\mathcal{C}_m$	Cohort subgraph discovered from the patient graph via Leiden algorithm
M	Total number of discovered cohorts
$\mathbf{z}_m$	Centroid embedding of cohort $\mathcal{C}_m$ obtained by mean pooling
$\mathcal{Q}$	Graph modularity objective optimized during cohort discovery
$\mathbf{X}^{(q)}$	EHR record of the target patient
$\mathbf{h}^{(q)}$	Embedding of the target patient
$\mathcal{C}_{\text{ret}}^{(q)}$	Set of top-K_c cohorts retrieved for the target patient
K_c	Number of cohorts retrieved for the target patient
$\mathcal{V}_m^{(q)}$	Set of anchor nodes selected from cohort $\mathcal{C}_m$ for the target patient
K_a	Number of anchor nodes selected per retrieved cohort
$\mathcal{F}^{(q)}$	Search frontier consisting of candidate demonstration nodes for the target patient
$\mathcal{S}^{(q)}$	Current demonstration set constructed for the target patient
K	Maximum number of demonstrations
v_i	Candidate demonstration node corresponding to patient i
v^*	Selected demonstration node that maximizes marginal information gain
x_{test}	Unlabeled test query constructed from the target patient's EHR record
$H_\theta(x_{\text{test}} \mid \mathcal{S})$	Conditional entropy of the LLM's prediction on the test query given demonstration set $\mathcal{S}$
$\Delta H_\theta(v_i \mid \mathcal{S})$	Marginal information gain obtained by adding candidate v_i to demonstration set $\mathcal{S}$
θ	Parameters of the target LLM
$\mathcal{N}(v)$	Neighbor set of node v in the patient graph, used for frontier expansion

Table 10: Key Notations Used in *GraphWalker*

J Code and Data Availability

To support reproducibility and facilitate future research, we will publicly release the full implementation of *GraphWalker* along with the processed datasets used in our experiments upon publication.

K Computational Resources and Software Environment

All experiments were conducted on a server equipped with two NVIDIA H20 GPUs (96 GB memory each) and 503 GB system RAM, running Ubuntu 22.04.5 LTS. The software environment was based on Python 3.11.11 with Conda 23.5.2, and experiments were implemented using PyTorch 2.6.0, HuggingFace Transformers 4.51.3, and SpaCy 3.8.4, all with default settings unless otherwise specified.

L Use of Large Language Models

In this work, large language models (LLMs) were used in a supportive role to assist with language refinement and programming-related tasks, including improving clarity, grammatical correctness, and presentation quality, as well as providing high-level guidance for code structuring and debugging. All LLM-assisted outputs were carefully reviewed, verified, and, when necessary, revised by the authors prior to inclusion. The core research ideas, methodological design, experimental setup, and result analysis were conceived and carried out entirely by the authors. LLMs did not contribute to the formulation of scientific hypotheses, the design of the proposed methods, or the derivation of research conclusions.