Grounded by Experience: Generative Healthcare Prediction Augmented with Hierarchical Agentic Retrieval

Chuang Zhao, Hui Tang, Hongke Zhao, Xiaofang Zhou, *Fellow, IEEE*, Xiaomeng Li, *Senior Member, IEEE*

***Abstract*—Accurate healthcare prediction is critical for improving patient outcomes and reducing operational costs. Bolstered by growing reasoning capabilities, large language models (LLMs) offer a promising path to enhance healthcare predictions by drawing on their rich parametric knowledge. However, LLMs are prone to factual inaccuracies due to limitations in the reliability and coverage of their embedded knowledge. While retrieval-augmented generation (RAG) frameworks, such as GraphRAG and its variants, have been proposed to mitigate these issues by incorporating external knowledge, they face two key challenges in the healthcare scenario: (1) identifying the clinical necessity to activate the retrieval mechanism, and (2) achieving synergy between the retriever and the generator to craft contextually appropriate retrievals. To address these challenges, we propose GHAR, a generative hierarchical agentic RAG framework that simultaneously resolves when to retrieve and how to optimize the collaboration between submodules in healthcare. Specifically, for the first challenge, we design a dual-agent architecture comprising Agent-Top and Agent-Low. Agent-Top acts as the primary physician, iteratively deciding whether to rely on parametric knowledge or to initiate retrieval, while Agent-Low acts as the consulting service, summarising all task-relevant knowledge once retrieval was triggered. To tackle the second challenge, we innovatively unify the optimization of both agents within a formal Markov Decision Process, designing diverse rewards to align their shared goal of accurate prediction while preserving their distinct roles. Extensive experiments on three benchmark datasets across three popular tasks demonstrate our superiority over state-of-the-art baselines, highlighting the potential of hierarchical agentic RAG in advancing healthcare systems.**

***Index Terms*—Retrieval augment generation, Agent collaboration, Healthcare prediction**

I. Introduction

HEALTHCARE is a pivotal domain for societal well-being, fundamentally influencing patient outcomes and the operational effectiveness of medical systems [1]–[3]. Prevailing healthcare prediction approaches can be broadly classified into discriminative and generative paradigms [4], [5]. Discriminative approaches, which have predominated in prior research, excel at capturing temporal patterns [6] and modeling higher-order co-occurrences among Electronic Health Record (EHR) [7]. However, these EHR ID-based methods inherently neglect entity semantics, thereby failing to uncover the underlying mechanisms beyond statistical associations and ultimately limiting model generalizability and interpretability. Recently, the brilliance of large language models (LLMs) in information retrieval has inspired researchers to explore their applications in healthcare [8]–[10]. Typical approaches involve utilizing LLMs to construct external knowledge bases [11]–[13] or employing them as foundation models for instruction tuning [14], [15], aiming to leverage their extensive parametric knowledge to enhance prediction accuracy, as illustrated in Fig. 1(a). Despite their potential, LLMs are prone to generating plausible yet factually inaccurate content, a phenomenon known as "hallucination" [16], [17]. For instance, within the context of medical dialogue, an LLM may erroneously validate a patient's self-diagnosis of "adrenal fatigue"—a condition not recognized by mainstream endocrinology—by fabricating non-existent diagnostic criteria or citing spurious sources to substantiate its claim.

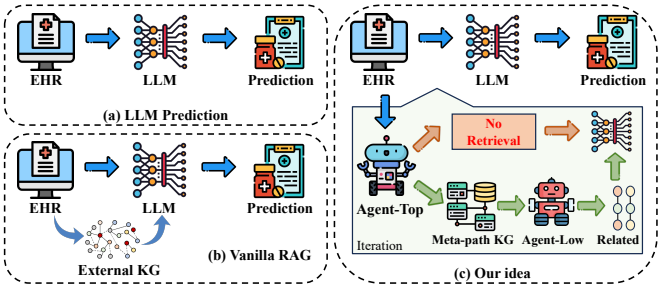


Fig. 1. Motivation Difference. (a) Forecasting using only LLM parameterized knowledge. (b) Single-round retrieve augmented generation with an external knowledge graph (KG). (c) Our idea utilizes hierarchical agents for iterative generation.

As illustrated in Fig. 1(b), recent generative initiatives like GraphRAG aim to minimize hallucinations through knowledge-intensive retrieval-augmented generation (RAG), typically employing a single-round retrieval to gather relevant evidence from external sources [18], [19]. While effective for straightforward tasks, this one-shot approach falls short in healthcare, where composite clinical queries demand intricate reasoning. Building on this, several RAG-based frameworks have been tailored for medical applications. However, they still encounter two primary limitations. First, systems like

C. Zhao, H Tang, and X. Li are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR, China; (e-mail: czhaobo@connect.ust.hk, eehtang@ust.hk, eexmli@ust.hk). X. Li is the corresponding author.

H. Zhao is with the College of Management and Economics, Laboratory of Computation and Analytics of Complex Management Systems (CACMS), Tianjin University, Tianjin 30072, China; (e-mail: hongke@tju.edu.cn)

X. Zhou is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR, China; (e-mail: zxf@ust.hk).

MedRAG [20] and KARE [8] either overlook or utilize static rules to determine the necessity of retrieval, failing to dynamically assess the clinical necessity for additional information. For example, a query related to a common chronic condition such as diabetes may not routinely require additional retrieval, whereas diagnosing a rare genetic disorder could benefit significantly from consulting specialized databases. Over-reliance on unnecessary retrieval may introduce extraneous noise, compromise output quality, and increase inference latency, as demonstrated in Fig. 9. Second, the retriever and the generator—the two core components of the RAG pipeline—often fail to synergize effectively. Conventional paradigms treat the training of retrievers and LLM generators as isolated processes [8], [21], where the former are typically optimized for document ranking metrics, such as Normalized Discounted Cumulative Gain (NDCG), while the latter are trained for conditional text generation [22]. This semantic incongruity adversely hampers their collaboration in healthcare. The generator, unaware of the retriever's selection rationale, tends to process retrieved evidence passively, leading to a "loss-in-the-middle" phenomenon where crucial clinical cues are obscured by non-actionable information, compromising accuracy. Consequently, as illustrated in Fig. 10(a), while KARE's extracted information is semantically related, it often fails to provide the precise evidence required for reliable clinical decision-making, leading to erroneous predictions.

To address these challenges, we propose GHAR, a novel generative framework designed to enhance RAG performance in healthcare through a hierarchical agentic retrieval architecture. First, to tackle the "when to retrieve" dilemma, we introduce a dual-agent architecture comprising Agent-Top and Agent-Low, as depicted in Fig. 1(c). Agent-Top serves as the primary physician, assessing the need for a deeper workup and initiating additional consultations, while Agent-Low functions as the consulting service that synthesizes the retrieved information into a cohesive summary of relevant clinical knowledge. Meanwhile, to achieve a more fine-grained narrowing of the retrieval scope, we introduce the critical concept of meta-path [23], [24] within heterogeneous graphs for partition awareness. Each meta-path (e.g., *disease*→*treated_by*→*drug*) represents a distinct type of clinical relationship. Once retrieval is deemed necessary, Agent-Top selects the appropriate meta-paths for coarse-grained identification of knowledge gaps, much like a physician identifies primary protocols before conducting a detailed consultation. This targeted approach extracts a focused, clinically pertinent data subset, sharpening retrieval precision for medical decision-making, as evidenced in Table VII and Fig. 8(c). Second, to unify retriever and LLM generator under a collaborative optimization framework, we formalize the healthcare decision-making as a Markov Decision Process (MDP) [25], [26], treating each component as a learning agent. We optimize these agents jointly through multi-agent reinforcement learning (RL) [27], [28], facilitating coordinated decision-making that enhances overall system performance. This approach aligns the objectives of LLM reasoning and retrieval, ensuring they function cohesively to produce accurate and well-supported clinical outputs. Furthermore, we design a diverse set of rewards tailored to the unique characteristics and commonalities of each agent. Our reward structure emphasizes lower overall costs, higher task accuracy, and the rationality of the reasoning path. Through continuous exploration in optimization, the model progressively identifies optimal strategies, allowing each agent to fulfill its designated role while promoting strong synergy among them.

To summarize, the contributions of this work are threefold:

- We are the first to devise a hierarchical agentic RAG tailored for healthcare prediction. GHAR not only resolves the critical issue of "when to retrieve" but also enables "collaborative optimization of the generator and retriever", establishing an effective and efficient RAG pipeline.
- We innovatively formulate the RAG optimization as a MDP process, employing multi-agent RL to synchronize the reasoning process and partition-aware retrieval within a unified framework. This enables nuanced decision-making, with varied rewards ensuring semantic relevance and synergy among agents.
- Extensive experimental results demonstrate the superiority of GHAR over state-of-the-art baselines.

II. Related Work

In this section, we review the closely related work, highlighting both connections and distinctions.

A. Healthcare Prediction

Healthcare prediction is a critical component in advancing medical practice, significantly influencing patient outcomes and operational efficiency of healthcare systems [3], [5], [29].

Data-driven healthcare prediction falls into two categories: discriminative and generative [8], [30]. Discriminative methods can be further divided into instance-based, longitudinal-based, and hybrid approaches. Instance-based models [7] often formulate graphs to depict relationships among entities in single-round EHRs; however, they are limited by their static nature. Longitudinal methods [6], [31], [32], on the other hand, account for sequential patterns and utilize RNN and Transformer in predicting patient outcomes. Despite these advancements, both paradigms remain predominantly ID-centric co-occurrence, prompting the development of hybrid approaches that enhance the utilization of external knowledge. Common strategies [12], [33] involve constructing external knowledge graphs or hypergraphs to facilitate the retrieval of relevant subgraphs for each patient visit. Recently, validations of LLMs in information retrieval have prompted researchers to explore their potential for generating additional corpora and supporting generative tasks [34]. Their typical approach involves utilizing LLMs' parametric knowledge to enrich external corpora or employing instruction learning to adapt general LLMs to the healthcare domain [35], [36]. Nonetheless, challenges remain, particularly regarding LLM hallucinations, which have not been effectively addressed [18]. Although frameworks like KARE [8] and MedRAG [20] attempt to integrate RAG pipelines for factual verification, their approach suffers from two key limitations. Not only does their indiscriminate reliance on single-round retrieval incur significant costs, but their

isolated optimization process also risks creating an alignment gap between the LLM generation and the retrieval process.

Our approach is grounded in the generative genre. Unlike hybrid discriminative methods, we directly use LLMs as the backbone, allowing us to capture semantic nuances beyond small databases and achieve enhanced decision-making capabilities. In contrast to the existing generative approaches, we are the first to apply agentic RAG and iterative deep thinking specifically to healthcare prediction, distinguishing our approach from existing single-round RAG models like KARE and MedRAG. Furthermore, our retrieval process is aligned with overarching healthcare objectives and is optimizable—a feature that is currently lacking in existing work.

B. Retrieval Augment Generation

RAG systems mitigate LLM limitations, such as factual hallucinations and outdated knowledge, by retrieving relevant information from external corpora [37], [38]. This approach is particularly crucial in knowledge-intensive fields like healthcare [39], where accuracy and factual evidence are vital for effective clinical decision-making.

The earliest implementation, Naive RAG [40], relies on traditional retrieval techniques such as TF-IDF and BM25 to fetch documents from static datasets. While effective for keyword-based queries, Naive RAG struggles to capture semantic nuances, limiting its contextual understanding. In contrast, Advanced RAG [41], [42] leverages dense vector search models, like dense passage retrieval, and neural ranking algorithms to encode queries and documents into high-dimensional vector spaces. This enables better semantic alignment and retrieval accuracy, which is crucial in healthcare settings where nuanced understanding can significantly influence patient outcomes. Modular RAG [43] further enhances flexibility by decomposing the retrieval and generation pipeline into independent, reusable components, allowing for domain-specific customization tailored to varied healthcare applications. Building on these concepts, Graph RAG [19], [44], [45] incorporates graph-structured data to enhance multi-hop reasoning and contextual richness, making it particularly effective in fields like healthcare and law, where complex relationships between entities are vital. The most recent advancement, Agentic RAG [21], [46], introduces autonomous agents capable of dynamic decision-making and workflow optimization. This paradigm employs query refinement and adaptive retrieval strategies to address complex, real-time, multi-domain queries.

Our work synthesizes elements from both Graph RAG and Agentic RAG into a cohesive framework. Unlike existing RAG systems that perform retrieval for all queries, our approach employs an iterative reasoning process to determine the optimal timing for retrieving external knowledge. We also introduce meta-paths as innovative partitioning strategies for fine-grained and efficient retrieval. Additionally, we formalize the interactions between agents as a Markov Decision Process (MDP), enabling us to unify the optimization of Agent-Top (which determines when to retrieve) and Agent-Low (which handles task-relevant retrieval) under a shared reward mechanism. Table I outlines the key distinctions between our algorithm and existing generative approaches.

TABLE I
KEY DIFFERENCE WITH GENERATIVE BASELINES. ALL REFERS TO SEARCHING IN THE ENTIRE KNOWLEDGE BASE, AND PARTIALLY REFERS TO SEARCHING IN THE SELECTED META-PATH KNOWLEDGE PARTITION. NA REFERS TO NOT APPLICABLE.

Medthod	External Knowledge	Knowledge Used	Training	LLM-Retriever Optimization	Iteration	Reward
LightRAG [19]	KG	All	SFT	Separated	Single-round	NA
Search-R1 [21]	Document	All	RL	Separated	Multi-round	Single
MedRAG [20]	KG	All	SFT	Separated	Single-round	NA
KARE [8]	KG	All	SFT	Separated	Single-round	NA
Medical-SFT [47]	NA	NA	SFT	NA	Single-round	NA
Ours	KG	Partially, (Meta-path)	SFT, RL	Unified	Multi-round	Diverse

III. PROPOSED METHOD

We first introduce the preliminaries, then provide an overview of the proposed GHAR, and detail submodules.

A. Preliminaries

Healthcare Dataset: Each patient's medical history is represented as a sequence of visits, denoted as $\mathcal{U}^{(i)} = (\mathrm{u}_1^{(i)}, \mathrm{u}_2^{(i)}, \ldots, \mathrm{u}_{\mathcal{N}_i}^{(i)})$, where i identifies the i-th patient in the set $\mathcal{U}$ and $\mathcal{N}_i$ indicates the total number of visits. For clarity, we use a single patient as an example and omit the superscript i. Each visit yields multiple views of data, represented as $\mathrm{u}_j = \{\mathrm{d}_j, \mathrm{p}_j, \mathrm{m}_j\}$, where diagnosis ($\mathrm{d} \in \mathcal{D}$), procedure ($\mathrm{p} \in \mathcal{P}$), and medication ($\mathrm{m} \in \mathcal{M}$), each represented as a set respectively. For instance, in the j-th visit, d_j, may include {Heart Failure, Diabetes}, indicating that the patient has two serious diseases.

Biomedical Knowledge Graph: Our KG corpora is represented as a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes (entities) and $\mathcal{E}$ denotes the set of edges (relationships). We further use $\tilde{\mathcal{V}}$ to denote the node type. The relationships are primarily sourced from PrimeKG [12], [48], a popular and comprehensive knowledge base for healthcare. Its primary data sources include public knowledge graphs (e.g., DBpedia, Wikidata), academic literature, and medical databases, ensuring a rich and accurate knowledge graph. Additionally, following [23], [24], we define meta-paths as sequences of node types and edges that capture specific relationships, represented as $o = (\tilde{v}_i, e_{ij}, \tilde{v}_j)$, where $\tilde{v} \in \tilde{\mathcal{V}}$ and $e_i \in \mathcal{E}$. The set $\mathcal{O}_{\text{meta}}$ refers to the universal meta-path set, i.e., $\mathcal{O}_{\text{meta}} = \{o_1, \cdots, o_{\mathcal{X}}\}$, where $\mathcal{X}$ denotes the total number of meta-paths within PrimeKG.

Parametric Knowledge & RAG: Parametric knowledge can be formalized as $\mathrm{llm}(f_*(\mathrm{u}_i))$, where $f_*(\cdot)$ denotes the prompt template. Conversely, RAG can be formalized as $\mathrm{rag}(f_*(\mathrm{u}_i, \mathrm{r}_i))$, where r_i denotes the retrieved relevant content. The core difference between the two approaches lies in whether the retrieval process is invoked.

Task Formulation: Based on the literature [8], [11], [12], we present the formal definitions for three common tasks in healthcare prediction. Formally,

- **24h-Decompensation (DEC Pred)** focuses on a binary classification task aimed at predicting the likelihood of a patient's decompensation risks. It analyzes $[\mathrm{u}_1, ..., \mathrm{u}_{j+1}]$ to determine the decompensation risk within 24 hours, where the label $\mathbf{y}[\mathrm{u}_{j+1}] \in \mathbb{R}^{1\times 2}$, indicating either yes or no.

TABLE II
MATHEMATICAL NOTATIONS.

Notations	Descriptions
$\mathcal{U}$	any ehr dataset
u	patient u's ehr data
$\mathcal{D}$, $\mathcal{P}$, $\mathcal{M}$	diagnosis, procedure, and medication set
d, p, m	multi-hot code of diagnosis, procedure, and medication
$\mathcal{G}$	biomedical KG
$\mathcal{V}, \mathcal{E}$	node set, edge set
$\tilde{\mathcal{V}}, \tilde{\mathcal{E}}$	node type, edge type
$\mathcal{O}_{\text{meta}}, \tilde{\mathcal{O}}_{\text{meta}}$	all meta-paths, generated meta-paths
$f_*(\cdot)$	prompt template
$\mathcal{H}_{\text{rea}}$	accumulated reason history
$\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}$	state set, action set, transition probability matrix, rewards
q, $\mathcal{Q}_{\text{sub}}$	query, query queue
$\text{llm}(\cdot), \text{rag}(\cdot)$	llm generation, rag generation
$\Phi(N)$	Top-N retrieval
$\mathcal{I}$	the number of max iteration
y, $\hat{\text{y}}$	ground-truth, prediction
η	trade-off weight
$\mathbb{E}$	mathematics expectations
$\mathbb{I}$	format examination
A	advantage
θ, ψ	actor, critic parameters

- **Readmission Prediction (READ Pred)** involves a binary classification task aimed at forecasting the probability of a patient being readmitted within a certain period after discharge. It strives to predict $\text{y}[\phi(\text{u}_{j+1}) - \phi(\text{u}_j)]$ using $\{\text{u}_1, \text{u}_2, \ldots, \text{u}_j\}$. Here, $\phi(\text{u}_j)$ signifies the encounter time for visit u_j. The outcome $\text{y}[\phi(\text{u}_{j+1}) - \phi(\text{u}_j)]$ is assigned 1 if $\phi(\text{u}_{j+1}) - \phi(\text{u}_j) \leq \kappa$, otherwise 0. In this study, $\kappa = 15$, aligning with [7], [11].
- **Length-of-stay Prediction (LOS Pred)** is formulated as a multi-class classification problem to categorize the predicted length of stay into intervals. The aim is to forecast the length of ICU stay at time $j+1$, based on EHR data $\{\text{u}_1, ..., \text{u}_{j+1}\}$. We set intervals $\mathcal{C}$ into 10 splits following [11], [30], i.e., $\text{y}[\text{u}_{j+1}] \in \mathbb{R}^{1\times|\mathcal{C}|}$.

Key mathematical symbols of this paper are listed in Table II.

Solution Overview. Our solution aims to identify the optimal moment to activate the retrieval module and collaboratively optimize sub-modules to craft contextually appropriate retrievals. We first define an **Agent-Top**, which is responsible for high-level decisions regarding whether to engage in deeper reasoning and when to trigger the RAG or the LLM. This decision-making process is crucial, as it determines how efficiently the system can respond to dynamic patient needs. Upon invoking RAG, Agent-Top dynamically selects relevant meta-paths and transmits them to **Agent-Low** for focused attention. These meta-paths greatly narrow the knowledge base and offer a coarse-grained preliminary screening of information. Agent-Low engages directly with the externally extracted knowledge to generate precise and task-relevant responses, serving as contextual input for the historical reasoning paths in subsequent iterations. The optimization process for the two agents is structured as a unified MDP, supported by diverse rewards to ensure the distinct roles of each agent while aligning with shared task objectives. This strategy collaboratively optimizes the retrieval and generation processes, fostering a bidirectional demand-aware environment. The methodological overview is illustrated in Fig. 2.

B. Overview of the MDP Modeling

We first overview the MDP [25], [26] for the construction of the Agent-Top and Agent-Low. This MDP framework captures the interactions and decision-making processes of both agents. Formally,

$$\mathbf{MDP} = \left\langle \mathbf{S}^T \times \mathbf{S}^L, \mathbf{A}^T \times \mathbf{A}^L, \mathbf{P}, \mathbf{R}^T \oplus \mathbf{R}^L \right\rangle, \quad (1)$$

where $\mathbf{S}^T$ and $\mathbf{S}^L$ are the sets of states for Agent-Top and Agent-Low, respectively. $\mathbf{A}^T$ and $\mathbf{A}^L$ denote their action sets, while $\mathbf{R}^T$ and $\mathbf{R}^L$ represent their rewards. $\mathbf{P}$ denotes the transition probability matrix. Then, we present the state transition for the unified MDP model. Formally, for the state of $t+1$-th iteration,

$$\begin{aligned} s_{t+1} = \left(s_{t+1}^T, s_{t+1}^L\right) &\sim \mathbf{P}\left(\cdot \mid s_t^T, s_t^L, a_t^T, a_t^L\right) \\ &\sim \mathbf{P}^T(\cdot|s_t^T, a_t^T) \times \mathbf{P}^L(\cdot|s_t^L, a_t^L, \xi(s_t^T)), \end{aligned} \quad (2)$$

where a_t^T and a_t^L are the actions taken by the Agent-Top and Agent-Low, respectively. ξ denotes hierarchical state transfer into low space. It can be observed that the two agents mutually influence each other, collectively determining the trajectory of the overall state. This interaction provides a rationale for our motivation in pursuing collaborative optimization. The objective of the MDP is to maximize the expected cumulative reward [26] by selecting an optimal policy π, expressed as:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(\mathbf{R}^T(s_t, a_t^T) + \mathbf{R}^L(s_t, a_t^L)\right)\right], \quad (3)$$

where γ is the discount factor that balances current and future rewards. The policy π is defined as a mapping from states to actions, determining the best action to take in a given state.

C. Personalized Prompt Collection

For a healthcare prediction task, we first process the input query based on task requirements and patient information,

$$\text{q}_{\text{u}}^0 = f_{\texttt{query}}(\mathcal{T}, \text{u}), \quad (4)$$

where $f_*(\cdot)$ denotes the prompt template and $\mathcal{T}$ signifies the task information. Their details can be found in Section VII. Recognizing that a single prompt may introduce bias, we draw on [50], [51] and utilize query rewriting to generate multiple initial queries. Formally,

$$\text{Q}_{\text{sub}} = f_{\text{generate}}(\text{q}_{\text{u}}^0, K), \quad (5)$$

where Q_{sub} denotes the query set and K signifies the number of rewriting. Here we set $K = 3$ to maintain prompt diversity. Given that our approach is iterative, we utilize a queue to store Q_{sub}. More precisely, for the t-th iteration, we extract the query $\text{q}_{\text{sub}}^t \in \text{Q}_{\text{sub}}$ that entered the queue first to proceed with the next agent iteration.

D. Agent-Top: Coarse-grained Knowledge Navigation

Agent-Top acts as a high-level decision-maker, determining whether to engage in deep reflection (i.e., whether to terminate the current iteration) and whether to invoke the partition-aware retrieval process. Its state at iteration t is defined as,

$$s_t^T = (\text{q}_{\text{sub}}^t, \mathcal{H}_{\text{rea}}) \in \mathbf{S}^T, \quad (6)$$

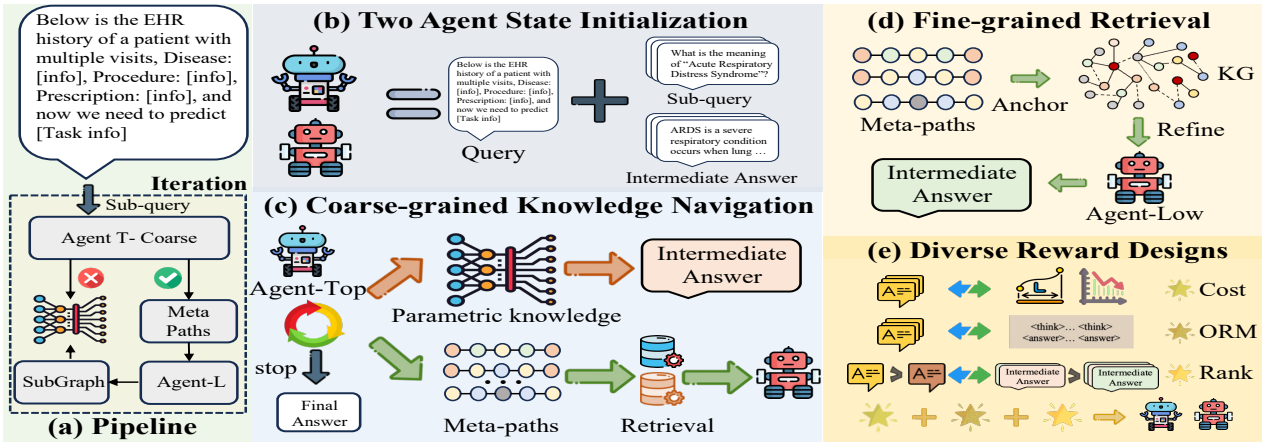


Fig. 2. Overview of *GHAR*. (a) Outline of the pipeline for each iteration, potentially involving LLM or LLM+RAG paths. (b) The agent's state is determined by the initial query and all historical reasoning paths. (c) For Agent-Top, it is essential to determine both whether to terminate the process and when to trigger retrieval. (d) For Agent-Low, it summarizes extracted external knowledge to produce a task-relevant response. (e) The diverse rewards design includes cost reduction, format standardization, and accuracy, as well as ranking, to maintain the role distinction and collaborative dynamics between the two agents. ORM denotes the outcome-supervised reward [49].

where $\mathrm{q}_{\text{sub}}^t$ is the current sub-query being processed. $\mathcal{H}_{\text{rea}}$ represents the current accumulated reasoning history. Specifically, for the first subquery, $\mathcal{H}_{\text{rea}}$ is an empty string " ", and subsequent entries are updated by appending all previous subqueries and their corresponding intermediate answers, as depicted in Fig. 2(b). The action space of Agent-Top includes two-step actions, formally,

$$a_t^{T,1} = \begin{cases} \texttt{llm}_{\text{top}}(f_{\text{llm}}(s_t^T)), \\ \texttt{rag}(f_{\text{meta}}(s_t^T, \mathcal{O}_{\text{meta}})), \end{cases} \quad a_t^{T,2} = \begin{cases} \text{terminate}, \\ \text{continue}, \end{cases} \tag{7}$$

where $a^{T,1} \in \mathbf{A}^T$ determines whether to invoke LLM or RAG. If LLM is selected, it directly utilizes its parametric knowledge for response generation; conversely, if RAG is chosen, the process transitions into the RAG framework rag(·). It is important to note that we do not perform direct retrieval from the entire external KGs. Instead, we focus on generating meta-path IDs $\tilde{\mathcal{O}}_{\text{meta}}$ and subsequently employ matched meta-paths $\tilde{\mathcal{O}}_{\text{meta}} \cap \mathcal{O}_{\text{meta}}$ for GraphRAG into Agent-Low, facilitating a fine-grained RAG process. This approach mitigates the computational load by constraining semantic neighborhoods within structured relational frameworks, thereby enabling models to focus on semantically relevant subspaces aligned with task-specific meta-paths, as evidenced in Table VII and Fig. 8(c). The parameter $a^{T,2} \in \mathbf{A}^T$ determines whether to generate sub-queries for deeper reasoning. A termination decision indicates that current knowledge suffices for query response. Otherwise, a new subquery will be generated and added to Q_{sub} to bridge the knowledge gap. This simulates the human process of incremental thinking, gradually unraveling to find the final answer to the problem, as detailed in Eq 13. The reward function for Agent-Top is designed to promote reasoning efficiency and path efficacy. Formally,

$$\begin{aligned} \mathbf{R}_{\text{reason}}^T &= 1 - \left| \frac{\text{len}(\mathcal{H}_{\text{rea}})}{L} - 1 \right|, \\ \mathbf{R}_{\text{path}}^T &= |\tilde{\mathcal{O}}_c| - 0.5 \cdot |\tilde{\mathcal{O}}_e| - 0.5 \cdot |\tilde{\mathcal{O}}_r|, \end{aligned} \tag{8}$$

where $\mathbf{R}_{\text{reason}}^T$ evaluates the overall length of the reasoning chain, while $\mathbf{R}_{\text{path}}^T$ measures the format of integrated meta-paths. L denotes the expected reason length and is set to 3. $|\tilde{\mathcal{O}}_c|, |\tilde{\mathcal{O}}_e|, |\tilde{\mathcal{O}}_r|$ denote the counts of correct IDs, erroneous IDs, and duplicated IDs within $\tilde{\mathcal{O}}_{\text{meta}}$, respectively. Please note that to ensure the rewards are on the same scale, we follow [52] for normalization. We encourage shorter chains and effective meta-paths generation, while ensuring performance, as this can reduce the computational burden during the RAG process.

E. Agent-Low: Fine-grained Retrieval

Once Agent-Top decides to trigger the RAG process, the selected meta-paths and subqueries will be transferred to the next stage for processing. The state of Agent-Low is formulated as,

$$s_t^L = (\mathrm{q}_{\text{sub}}^t, \mathcal{H}_{\text{rea}}, \mathcal{G}_{\text{sub}}) \in \mathbf{S}^L, \tag{9}$$

where $\mathcal{G}_{\text{sub}}$ is retrieved based on the generated meta-paths in Eq. 7. Specifically, we first execute the GraphRAG process for each meta-path partition to extract relevant nodes and edges, thereby constructing the necessary retrieved corpora. This subgraph will then be collaboratively passed to Agent-Low to build the state. Formally,

$$\mathcal{G}_{\text{sub}} = \bigcup_{\tilde{\mathcal{O}}_c}^{i} \{\Phi_{\text{nodes}}(\mathrm{q}_{\text{sub}}^t, \mathcal{G}_i, N) \cup \Phi_{\text{edges}}(\mathrm{q}_{\text{sub}}^t, \mathcal{G}_i, N)\}, \tag{10}$$

where $\mathcal{G}_{\text{sub}}$ denotes the retrieved corpus, $\mathcal{G}_i$ denotes the KG subgraph under meta-path i, and $\Phi(\cdot)$ refers to a specific retrieval algorithm. N refers to the Top-N retrieval in GraphRAG. Here, we choose E5 [53] as the retriever $\Phi(\cdot)$, and we also present modules for other retrieval algorithms in section V-B, which are flexible. Then, the action of Agent-Low is to further summarize these KG subgraphs, generating the task-relevant response as the intermediate answer for the RAG phase. Formally,

$$a_t^L = \text{llm}_{\text{low}}(f_{\text{rag}}(s_t^L)) \in \mathbf{A}^L, \tag{11}$$

where llm_{low} shares the same LLM with llm_{top} for convenience and to ensure semantic consistency. The reward function for Agent-Low incentivizes concise and relevant retrievals,

$$\mathbf{R}_{\text{rel}}^L = \text{Sim}(a_t^L, \mathrm{q}_{\text{sub}}^t) + \text{Sim}(a_t^L, \mathcal{G}_{\text{sub}}), \tag{12}$$

where $\mathbf{R}_{\text{rel}}$ evaluates how well the retrieved information addresses the sub-query, while also minimizing deviation from the original evidence. Sim$(\cdot)$ denotes the normalized token overlap count, a widely adopted metric in existing LLM research [27], [28].

F. Prediction & MDP Optimization

Iterative Reasoning. We adopt an iterative approach to deepen the reasoning process. Specifically, after each subquery receives an intermediate answer, Agent-Top will decide whether to directly use the current information to generate the ultimate answer $\hat{\text{y}}$, as determined by $a^{T,2}$ in Eq. 2, or to formulate a deeper subquery to bridge the knowledge gap. Formally,

$$\hat{\text{y}} = \text{llm}_{\text{top}}(\text{q}_{\text{u}}^0, \mathcal{H}_{\text{rea}}), \quad \text{if } a^{T,2} = \text{terminate}. \tag{13}$$

If $a^{T,2}$ is continue, $\text{Q}_{\text{sub}} \leftarrow \text{Q}_{\text{sub}} \cup \text{llm}_{\text{top}}(f_{\text{sub}}(\text{q}_{\text{sub}}^t, \mathcal{H}_{\text{rea}}))$, referring to add a new subquery to the queue. If the maximum number of iterations $\mathcal{I}$ is reached, it will automatically return "terminate". According to this paradigm, the Agent-Top can adaptively determine whether deep thinking is necessary and decide when to terminate. Additionally, the queue data structure facilitates the model's comprehensive understanding of the essential knowledge required to address the problem, dynamically expanding the boundaries of its knowledge, as evidenced in Fig. 10(b).

Rewards Design & Optimization. In light of the limitations posed by the independence of previous optimizations for RAG and LLM, we unify the two agents into a single MDP process and optimize them using multi-agent PPO [52]. To ensure that both agents work collaboratively, we design a shared reward mechanism. Beyond the agent-specific rewards in Eq. 8 and 12, we introduce two additional rewards $\mathbf{R}_{\text{orm}}$ and $\mathbf{R}_{\text{rank}}$ that require collaboration between the agents. Formally,

$$\mathbf{R}_{\text{all}}(s_t, a_t) = \mathbf{R}_{\text{cost}} + \eta \mathbf{R}_{\text{orm}} + \mathbf{R}_{\text{rank}}, \tag{14}$$

$$\begin{cases} \mathbf{R}_{\text{cost}} = \mathbf{R}_{\text{reason}}^T + \mathbf{R}_{\text{path}}^T + \mathbf{R}_{\text{rel}}^L, \\ \mathbf{R}_{\text{orm}} = \mathbb{I}(\text{y}, \hat{\text{y}}) + \mathbb{I}(\text{format}, \hat{\text{y}}) + \mathbb{I}(\text{format}, a_t), \\ \mathbf{R}_{\text{rank}} = \max(\alpha, \text{Sim}(\mathcal{H}_{\text{rea}}, \mathcal{H}_{\text{rea}}^{\text{pos}}) - \text{Sim}(\mathcal{H}_{\text{rea}}, \mathcal{H}_{\text{rea}}^{\text{neg}})), \end{cases} \tag{15}$$

where $\mathbf{R}_{\text{orm}}$ denotes the degree of accuracy and format matching with the final answer, i.e., outcome-supervised reward [49]. We provide additional weight control η to balance the trade-off between $\mathbf{R}_{\text{orm}}$ and other auxiliary rewards. $\mathbf{R}_{\text{rank}}$ is used to maintain the distance α between the correct reasoning paths and the negative reasoning paths. $\mathcal{H}_{\text{rea}}^*$ refers to the exploration reasoning path used as a reference. More specifically, prior to training, we follow prior work [27], [38], [49] in using a stronger LLM (Qwen2.5-7B) [47] to generate reasoning rollouts. The common practice is to select correctly answered queries and use their reasoning paths as positive examples ($\mathcal{H}_{\text{rea}}^{\text{pos}}$) for alignment. Unlike them, our approach diverges by additionally recording the model's opposite decision points (e.g., the choice of whether to perform retrieval) and incorporating them into a negative set ($\mathcal{H}_{\text{rea}}^{\text{neg}}$), thereby constructing a pairwise loss. We aim to simultaneously imitate the reasoning process of the correct paths and steer away from the incorrect reasoning paths.

In common multi-agent optimization scenarios, the actor-critic architecture [52], [54] is widely employed, taking into account the reference model(θ_{ref}), actor model(θ), and critic model(ψ). Both agents share parameters using Qwen2.5-3B as the backbone for efficiency, with further possibilities discussed in section V-B. The designed multi-agent PPO loss is as follows,

$$\mathcal{L}_{\text{actor}}(\theta) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} \left[\min \left(r_\theta A_t, \text{clip}\left(r_\theta, 1-\epsilon, 1+\epsilon\right) A_t \right) \right], \tag{16}$$

where $r_\theta = \frac{\pi_\theta(a_t^T, a_t^L | s_t)}{\pi_{\theta_{\text{ref}}}(a_t^T, a_t^L | s_t)}$ and $\pi_{\theta_{\text{ref}}}$ is a reference policy. $A_t = \sum_{t=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$ denotes the advantage function, where δ_{t+l} refer to the TD-error [52]. Correspondingly, we present the optimization for the critic network. Formally,

$$\mathcal{L}_{\text{critic}}(\psi) = \sum_t \left[(V_\psi(s_t) - \mathbf{R}_{\text{shared}}(s_t, a_t))^2 \right], \tag{17}$$

where V is the value estimation and $\mathcal{L}_{\text{critic}}(\psi)$ is used to ensure the stability of the critic network within the actor-critic structure. Finally, our complete multi-agent optimization loss combines actor loss and critic loss. Formally,

$$\mathcal{L}_{\text{multi-agent}}(\theta, \psi) = \mathcal{L}_{\text{actor}}(\theta) + \mathcal{L}_{\text{critic}}(\psi). \tag{18}$$

Our approach significantly differs from previous work that optimizes the LLM generation and retrieval process in isolation. This unified optimization approach ensures that both agents learn to collaborate effectively while maintaining their specialized roles in the healthcare prediction framework. The overall pipeline of our proposed GHAR can be seen in Algorithm 1.

Algorithm 1 The Algorithm of *GHAR*.

Input: EHRs $\mathcal{U}$, KG $\mathcal{G}$, Queue Q_{sub}, Max iteration $\mathcal{I}$;
Output: Policy parameters θ;

```
Personalized prompt generation in Eq. 4;
Prompt rewriting in Eq. 5;
while Q_sub ≠ ∅ & t ≤ I do
    Extract a query q_sub from Q_sub;
    Agent-Top initialization in Eq. 6;
    if a_t^{T,1} = llm then
        a_t^{T,1} = llm_top(f_llm(s_t^T));
    else if a_t^{T,1} = rag then
        # Agent-Top Meta-path Navigation
        a_t^{T,1} = rag(f_meta(s_t^T, O_meta));
        # Agent-Low RAG Process
        Agent-Low initialization in Eq. 10;
        Agent-Low action in Eq. 11;
    end if
    Termination examination in Eq. 7 & Eq. 13;
    Reward calculation in Eq. 14-15;
    Actor-critic loss in Eq. 18;
    Update the parameters θ, ψ;
    t = t + 1;
end while
return Policy parameters θ
```

IV. Experiments

Datasets & Baselines. We utilize three popular datasets for healthcare prediction: MIMIC-III [56], MIMIC-IV [57], and eICU [58]. The MIMIC-III database is a large, publicly available dataset of de-identified health data from over 40,000

TABLE III
DATA STATISTICS. DUE TO THE DIVERSE TIME REQUIREMENTS OF VARIOUS TASKS, WE USE THE LOS PRED TASK AS A REPRESENTATIVE EXAMPLE, GIVEN ITS LENIENT DATA RESTRICTIONS. # DENOTES THE NUMBER OF ITEMS, AND AVG REPRESENTS THE AVERAGE VALUE. FOLLOWING THE POPULAR DATA PARTITIONING PROTOCOLS [33], [55], WE DIVIDE THE DATASET INTO TRAINING, VALIDATION, AND TEST SETS USING A 6:2:2 RATIO.

Items	MIMIC-III	MIMIC-IV	eICU
# of patients / # of visits	35,707 / 44,399	46,178 / 154,962	8,600 / 18,691
diagnose. / procedure. / medication. set size	6,662 / 1,978 / 197	19,438 / 10,790 / 200	1,358 / 437 / 1,411
avg. # of visits	1.2434	3.3551	2.1734
avg. # of diagnose history per visit	17.7373	48.9516	14.1709
avg. # of procedure history per visit	6.1718	8.7626	45.9671
avg. # of medication history per visit	38.2772	82.7437	26.0229

critical care admissions for research in medical informatics. MIMIC-IV is a more complicated version from the U.S., encompassing over 70,000 admissions and providing more extensive longitudinal EHR histories. In contrast, the eICU dataset focuses on a diverse cohort of ICU patients across multiple hospitals, encompassing around 200,000 admissions and including detailed clinical data that supports various predictive modeling tasks. We follow [8], [55] for data processing, with statistics presented in Table III.

As detailed in section II-A, we select fifteen competitive baselines. For discriminative models, we choose approaches such as GRASP [7], StageNet [6], and SHAPE [32], all of which are ID-based methods designed to capture the collaborative signals among entities. We also introduce hybrid methods like GraphCare [11], EMERGE [12], FlexCare [59], and UDC [33], which enhance traditional discriminative methods using language models (LM-based). For generative models, we include several tuning-free LLMs, such as DeepSeekR1-7B [49], alongside medical LLMs like Meditron-7B [35] and BioMistral-7B [36], which are fine-tuned with medical corpora. Several tuning-required baselines from our scenario are also incorporated for comparison. Medical-SFT (Supervised Fine-Tuning) serves as an effective reasoning method [47] and is employed on the Qwen2.5-7B for a fair comparison. Additionally, we incorporate RAG methods such as Search-R1 [21], LightRAG [19], KARE [8], and MedRAG [20], with the latter two specifically designed for medical contexts. Given that our algorithm utilizes a 7B model during the warm-up phase, we ensure fairness in parametric knowledge by using a consistent LLM backbone for LoRA tuning (rank=8) across all generative models. For Search-R1 and GHAR, we employ the PPO variant for exploration.

Implementation Details. We implement GHAR and all baselines using PyTorch 3.11 and LangChain [1], with a learning rate and training epochs set to 1e-4 and 3, respectively. We conduct our experiments on a hardware configuration featuring a 12-core Intel Xeon CPU and eight NVIDIA A800 GPUs. In the initial phase, akin to [27], [38], [49], we first employ Qwen2.5-7B [47] for rejection sampling [49] on the training set to obtain warm-up samples. More precisely, we leverage the reasoning pathways of correctly answered samples from this more advanced LLM as warm-start data, ensuring that the model effectively learns the initial reasoning ability. Next, we conduct two phases of training: SFT ensures the model learns the basic format, while RL allows for further exploration to enhance performance. We set Agent-Top and Agent-Low to share the same LLM parameters at the start of training, specifically Qwen2.5-3B [47], to reduce the inference load. We select E5 [53] as the retriever and utilize FAISS [56] to create the index. We also explore additional backbones in section V-B to test their applicability. The max number of meta-paths $|\tilde{\mathcal{O}}_{\text{meta}}|$, Top-N recall N, and ORM weights η are configured to 3, 1, and 5, respectively, based on the hyperparameter search outlined in section V-H. To enhance our training efficiency, we utilize the Ray [2] framework for distributed training, as shown in section V-F. The evaluation metrics selected include B-Accuracy, Accuracy, and F1-score, sourced from [8], [11], [60], which demonstrate significant clinical relevance.

Overall Results. As demonstrated in Tables IV to VI, our model outperforms all baseline models across three tasks, particularly in performance metrics that consider both positive and negative instances. From an algorithmic perspective, general-purpose LLMs like DeepSeekR1 exhibit limited performance due to gaps in the training corpus, which hinder their effectiveness in clinical scenarios. Effective supervised Medical-SFT can significantly mitigate this issue; however, the reliance on static datasets in SFT restricts generalization capabilities across diverse contexts. In contrast, RAG's ability to integrate external knowledge sources enhances its contextual awareness and adaptability to various queries, as evidenced by KARE's performance on READ tasks. We further investigate these distinctions in Section V-C, particularly regarding out-of-distribution generalization scenarios, where adaptability is crucial. RAG-based baselines, such as KARE and GHAR, outperform most ID-based baselines on the DEC (MIMIC-IV) task. This advantage arises from KARE and GHAR's capacity to leverage a sophisticated understanding of language and external knowledge, enabling them to capture complex semantic relationships and contextual nuances. In contrast, ID-based methods, such as GRASP, primarily focus on co-occurrence relationships, often neglecting the intricacies of language, which limits their performance. However, this observation is not universally applicable. In LOS Pred tasks, which are characterized by richer interactions, the performance gap narrows and may even reverse. This is likely due to the presence of more collaborative signals, which can lead to more definitive co-occurrence relationships.

In terms of task complexity, LOS predictions, along with the challenges of multi-class tasks and class imbalance, require a deeper understanding of label interactions and patient status comprehension. LM-based hybrid models, such as GraphCare and FlexCare, experience performance degradation in this context due to their reliance on discriminative approaches that depend on co-occurrence patterns. The noise inherent in frozen embedding processes can also introduce inaccuracies that negatively impact performance. Similarly, MedRAG and KARE, which do not optimize the retrieval process or employ selective retrieval, are likely to encounter similar retrieval noise during their generation process, ultimately leading to loss-in-middle. Moreover, we observe that as the training

[1]https://github.com/langchain-ai/langchain

[2]https://www.ray.io/

TABLE IV
PERFORMANCE COMPARISON: MIMIC-III DATASET. PLEASE NOTE THAT BOLD TEXT REPRESENTS OPTIMAL PERFORMANCE. B-ACCURACY REFERS TO BALANCED ACCURACY [55]. THE SYMBOL ↑ INDICATES THAT A HIGHER VALUE IS BETTER.

Genre	Task	24h-Decompensation		Readmission Prediction		Length-of-Stay Prediction	
	Method	Accuracy↑	B-Accuracy↑	Accuracy↑	B-Accuracy↑	Accuracy↑	F1-score↑
ID-based	GRASP [7]	0.9803	0.6858	0.5727	0.5710	0.4046	0.3271
	StageNet [6]	0.9779	0.7109	0.5938	0.5911	0.4091	0.3715
	SHAPE [32]	0.9769	0.8133	0.6009	0.6074	0.4245	0.3891
LM-based	GraphCare [11]	0.9826	0.7675	0.6084	0.6069	0.4212	0.3918
	EMERGE [12]	0.9804	0.7770	0.6144	0.6104	0.4189	0.3912
	FlexCare [59]	0.9816	0.8146	0.6185	0.6107	0.4320	0.3937
	UDC [33]	0.9828	0.8166	0.6139	0.6188	0.4357	0.3970
Generative-based	DeepSeekR1-7B [49]	0.1604	0.5379	0.5307	0.4906	0.1527	0.1366
	Meditron-7B [35]	0.2581	0.4813	0.5376	0.4932	0.0322	0.0141
	BioMistral-7B [36]	0.1078	0.5280	0.5463	0.5043	0.1109	0.0667
	LightRAG [19]	0.7721	0.4836	0.5564	0.5038	0.2860	0.1860
	Search-R1 [21]	0.9648	0.5561	0.5314	0.5117	0.3142	0.2583
	Medical-SFT [47]	0.9672	0.8145	0.5626	0.5279	0.4299	0.3957
	MedRAG [20]	0.9829	0.4984	0.5725	0.5269	0.2312	0.2045
	KARE [8]	0.9760	0.8114	0.5769	0.5156	0.3700	0.3427
	GHAR	**0.9842**	**0.8421**	**0.6244**	**0.6272**	**0.4486**	**0.4124**

TABLE V
PERFORMANCE COMPARISON: MIMIC-IV DATASET.

Genre	Task	24h-Decompensation		Readmission Prediction		Length-of-Stay Prediction	
	Method	Accuracy↑	B-Accuracy↑	Accuracy↑	B-Accuracy↑	Accuracy↑	F1-score↑
ID-based	GRASP [7]	0.9970	0.5995	0.6261	0.6263	0.3493	0.3273
	StageNet [6]	0.9911	0.6532	0.6324	0.6333	0.3467	0.3253
	SHAPE [32]	0.9968	0.6798	0.6325	0.6371	0.3751	0.3598
LM-based	GraphCare [11]	0.9968	0.7030	0.6327	0.6403	0.3735	0.3485
	EMERGE [12]	0.9962	0.6976	0.6316	0.6399	0.3859	0.3554
	FlexCare [59]	0.9960	0.7256	0.6282	0.6354	0.3658	0.3348
	UDC [33]	**0.9971**	0.7463	0.6388	0.6431	0.3916	0.3544
Generative-based	DeepSeekR1-7B [49]	0.1421	0.5436	0.5081	0.5026	0.0788	0.0510
	Meditron-7B [35]	0.1181	0.4848	0.5089	0.5025	0.0724	0.0245
	BioMistral-7B [36]	0.1225	0.4937	0.5092	0.5035	0.0814	0.0239
	LightRAG [19]	0.2582	0.6283	0.5040	0.4978	0.2129	0.1856
	Search-R1 [21]	0.9586	0.5295	0.5038	0.6020	0.2706	0.2819
	Medical-SFT [47]	0.9932	0.8110	0.6320	0.6335	0.3869	0.3564
	MedRAG [20]	0.9960	0.5705	0.5868	0.5683	0.2623	0.1701
	KARE [8]	0.9851	0.7978	0.6258	0.6446	0.3274	0.2865
	GHAR	0.9939	**0.8279**	**0.6473**	**0.6484**	**0.4035**	**0.3678**

TABLE VI
PERFORMANCE COMPARISON: EICU DATASET.

Genre	Task	24h-Decompensation		Readmission Prediction		Length-of-Stay Prediction	
	Method	Accuracy↑	B-Accuracy↑	Accuracy↑	B-Accuracy↑	Accuracy↑	F1-score↑
ID-based	GRASP [7]	0.9238	0.5114	0.8944	0.5575	0.2424	0.2200
	StageNet [6]	0.9699	0.5294	0.9119	0.5319	0.2646	0.2145
	SHAPE [32]	0.9633	0.5744	0.9132	0.5452	0.3081	0.2551
LM-based	GraphCare [11]	0.9712	0.5200	0.9026	0.5501	0.2948	0.2510
	EMERGE [12]	0.9703	0.5213	0.9167	0.5799	0.3088	0.2570
	FlexCare [59]	0.9716	0.5279	0.9073	0.5420	0.2982	0.2314
	UDC [33]	0.9711	0.5476	0.9169	0.5517	0.3091	0.2595
Generative-based	DeepSeekR1-7B [49]	0.1676	0.5360	0.6320	0.4625	0.0718	0.0456
	Meditron-7B [35]	0.0400	0.5037	0.9168	0.5027	0.2115	0.0905
	BioMistral-7B [36]	0.0698	0.5157	0.8990	0.5057	0.0857	0.0589
	LightRAG [19]	0.9468	0.4896	0.8990	0.4847	0.2243	0.1045
	Search-R1 [21]	0.9533	0.5006	0.9082	0.4999	0.1907	0.1186
	Medical-SFT [47]	0.9678	0.5183	0.9089	0.5009	0.2740	0.1950
	MedRAG [20]	0.9453	0.4989	0.9163	0.5014	0.2179	0.1373
	KARE [8]	0.9670	0.5178	0.9100	0.5107	0.2779	0.1773
	GHAR	**0.9720**	**0.5990**	**0.9171**	**0.5908**	**0.3200**	**0.2610**

dataset size increases, the performance of ID-based models gradually improves. For instance, on the MIMIC-IV dataset, ID-based models demonstrate enhanced performance due to a greater number of co-occurrence relationships, which aid in understanding patient states. Models like StageNet and SHAPE exhibit stronger performance, even surpassing Search-R1 and KARE on DEC and LOS tasks. In contrast, the impact on LLM-based baselines is minimal; their performance remains stable, as they rely on a comprehensive understanding of contextual relevance that is not solely dependent on co-occurrence signals. This resilience highlights the advantage of LLMs in managing diverse and complex datasets, enabling them to maintain performance even in smaller training scenarios, such as with the MIMIC-III dataset. We further explore out-of-distribution (OOD) reasoning in Section V-C to substantiate this point.

In summary, GHAR demonstrates flexibility in utilizing external knowledge, improving performance across diverse tasks and datasets.

V. MODEL ANALYSIS AND ROBUST TESTINGS

We conduct numerous robustness experiments to provide a more in-depth analysis. Without loss of generality, we use MIMIC-III for examination.

TABLE VII
ABLATION STUDY. NI REFERS TO THE USE OF A SINGLE-ROUND RATHER THAN AN ITERATIVE RAG. NT INDICATES THE ABSENCE OF AGENT-TOP. NL SIGNIFIES THE EXCLUSION OF AGENT-LOW. NM REFERS TO RETRIEVAL PERFORMED DIRECTLY OVER THE ENTIRE KNOWLEDGE GRAPH, AS OPPOSED TO EXTRACTION VIA META-PATH PARTITIONING. NS INDICATES THAT NO SHARED REWARD IS UTILIZED, RESULTING IN SEPARATE OPTIMIZATION FOR THE TWO AGENTS.

Algorithms	Metric	-NI	-NT	-NL	-NM	-NS	**GHAR**
DEC Pred	B-Accuracy	0.8295	0.8072	0.8151	0.8338	0.8099	**0.8421**
READ Pred	B-Accuracy	0.6221	0.6008	0.6169	0.6225	0.6096	**0.6272**
LOS Pred	F1-score	0.3928	0.3892	0.3804	0.4038	0.3919	**0.4124**

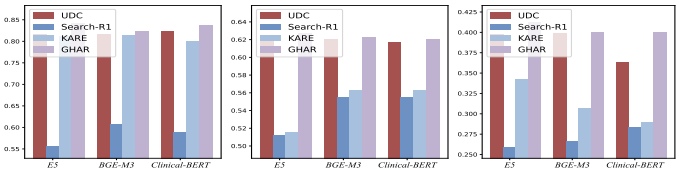
A. Ablation Studies

We conduct extensive ablation analyses on the designed submodules while keeping other components consistent to validate the effectiveness of each element. As shown in Table VII, each submodule is indispensable, as evidenced by the performance decline observed with any variant. GHAR-NI exhibits a performance degradation of over 2% in both DEC and LOS prediction tasks; the single-round retrieval lacks the depth of consideration necessary for effective outcomes. More precisely, it simply extracts external information based on the original query, losing the ability for a deep understanding of the knowledge gaps in the reasoning process. GHAR-NT and GHAR-NL, which respectively ablate Agent-Top and Agent-Low, rely solely on a single-agent role to complete tasks. The former approach degenerates into one that unconditionally pulls information from the entire KG for all queries, incurring a heavy computational burden and introducing noise by adding context even when RAG is not required. The latter crudely employs coarse-grained retrieval documents, potentially preventing the model from refining task-relevant knowledge, leading to a 4% performance drop (DEC Pred). GHAR-NM does not implement meta-path partitioning, directly extracting knowledge from medical entities across the entire knowledge graph. This straightforward Top-N extraction may be limited to coarse-grained knowledge structures, missing opportunities for finer-grained exploration. GHAR-NS eliminates shared rewards, optimizing the two agents separately. Our findings indicate that this approach results in a significant performance degradation of nearly 3%, primarily due to the lack of a unified objective among the agents. This misalignment leads to divergent semantic spaces between the LLM and the RAG components, thereby undermining their collaborative effectiveness. In summary, these experiments demonstrate the effectiveness of our submodules and provide deeper insights.

B. Diverse Retrievers & LLMs & Evaluation Settings

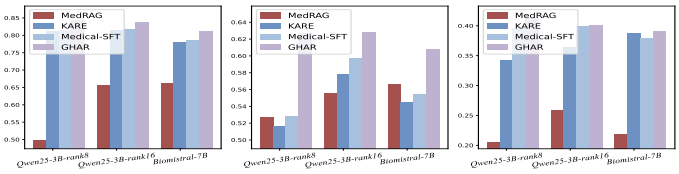
We replace various retrievers and LLM generators to assess the pluggability of our algorithm, as both play indispensable roles within our generative framework [5], [19].

Diverse Retrievers. Different retrieval models influence the nodes matched during the extraction of KGs, as variations in embedding spaces arise from distinct pretraining corpora and methodologies [33]. As depicted in Fig. 3, BGE-M3 [61] and Clinical-BERT [62] do not demonstrate significant gains. Despite BGE-M3 having more parameters and Clinical-BERT being fine-tuned for medical scenarios, we observe that the content extracted by both models shows only subtle differences. This may be due to significant language discrepancies among nodes during the KG matching process, as well as a gap between their pretraining corpora and healthcare scenarios. From another perspective, our Agent-Low can refine the retrieval content by focusing on subtle differences, enabling it to obtain information that is closely related to the task. This means it is less influenced by retrieval outputs when the differences between them are minimal.



(a) DEC Pred (B-ACC) (b) READ Pred (B-ACC) (c) LOS Pred (F1-score)

Fig. 3. Comparison under Diverse Retrievers. We employ the popular E5 [53], BGE-M3 [61], and Clinical-BERT [62].



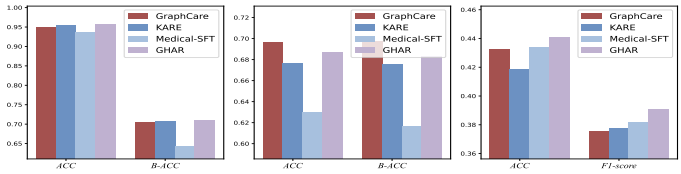
(a) DEC Pred (B-ACC) (b) READ Pred (B-ACC) (c) LOS Pred (F1-score)

Fig. 4. Comparison under Diverse LLMs. We employ Qwen2.5-3B with rank 8 (our method), Qwen2.5-3B with rank 16 [47], and BioMistral-7B [36].

Diverse LLM Backbones. We replace the LLM backbone (Qwen2.5-3B [47], with LoRA rank, 8) with larger parameter LLMs (Qwen2.5-3B [47], with higher-rank LoRA, 16) and medical LLMs (BioMistral-7B [36]) for model tuning. As shown in Fig. 4, larger parameter LLMs yield slight information gains on DEC and READ Pred. However, we observe that the improvements from the medical LLMs were not as pronounced as compared to the other backbone variants—in fact, their performance was at best comparable and at worst. This limited enhancement may arise from the overlap of knowledge within the medical LLMs and the external RAG knowledge, as both are based on training with open-source data. Additionally, the medical LLMs underwent extra reinforcement learning from human feedback specifically designed for textbook question-answering tasks, which may be less relevant to the healthcare predictions addressed in this study.

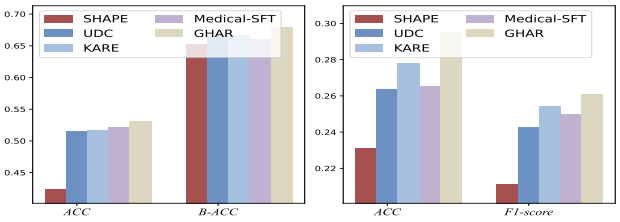
Diverse Experiment Settings. To demonstrate robustness across different training settings, we select a training data split of 0.8:0.1:0.1 for testing, following GraphCare [11] and KARE [8], as shown in Fig. 5. Please note that for generative patterns, AUROC and Balance ACC are equivalent, with the baseline derived from the generative pattern directly outputting text instead of logits. Comparisons with competitive baselines indicate that our performance improvements remain robust. We found that in situations with richer information, such as GraphCare in READ Pred, the discriminant still has an advantage, and does not completely lose its advantage as described in KARE. Our strategy improves accuracy through a greater number of cases, exposing agents to diverse patient patterns,

making it more robust than other generative approaches.



(a) DEC Pred (b) READ Pred (c) LOS Pred

Fig. 5. Diverse Training Settings. Please note that in KARE, DEC Pred denotes in-hospital mortality, not within 24h.



(a) MIMIC-III-REA→III-DEC (b) MIMIC-III-LOS→IV-LOS

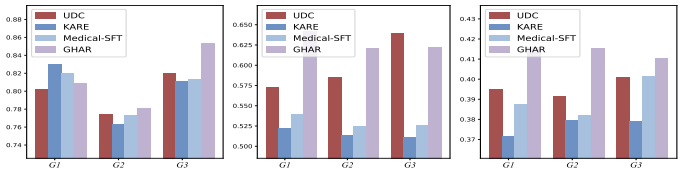
Fig. 6. OOD Examination. (a) cross-task scenario. (b) cross-dataset scenario. For both scenarios, we use one domain as the pre-training domain and then directly assess performance on the test set of the other domain.

C. Out-of-Distribution Examination

We conduct additional out-of-distribution (OOD) tests to verify the zero-shot transferability of our methods. This represents a significant advantage of generative approaches, as traditional discriminative methods adhere to the independent and identically distributed assumption, making it challenging to generalize to scenarios with significant distribution differences [37]. Specifically, we design two types of OOD scenarios: cross-task and cross-dataset. The former involves training on READ and testing on the DEC Pred task, while the latter consists of training on MIMIC-III and testing on MIMIC-IV. As shown in Fig. 6, both KARE and our method outperform SHAPE by more than 10%. This improvement is attributed to the generative paradigm's ability to capture the fundamental semantics of entities and labels, going beyond the shallow co-occurrence that discriminative models rely on for prediction, as further validated in Fig. 10(c). Moreover, in both tasks, our approach maintains an advantage over KARE and Medical-SFT, which can be attributed to the collaborative interplay between the two agents. This collaboration helps bridge the gap between retrieval and generation, allowing for an iterative selection process that accommodates a greater volume of external information. Meanwhile, RAG effectively addresses knowledge gaps in OOD scenarios by dynamically retrieving relevant and essential information from external knowledge bases. In contrast, Medical-SFT relies on fixed training data, limiting its generalization capability. As a result, GHAR demonstrates superior performance in OOD situations.

D. Group-Wise Performance

In real medical scenarios, the types of diseases among patients are often imbalanced, with common illnesses dominating the dataset [33]. Following [3], [33], we categorize



(a) DEC Pred (B-ACC) (b) READ Pred (B-ACC) (c) LOS Pred (F1-score)

Fig. 7. Group-wise Analysis. G1-G3 refer to groups based on disease rarity:G1: [0, 1/3), G2: [1/3, 2/3), G3: [2/3, 1].

patients based on the rarity of their diseases into groups G1-G3, with G1 representing the rarest group. As shown in Fig. 7, the overall performance of both discriminative and generative methods improves with the prevalence of diseases (G2→G3). For the discriminative methods, this improvement is likely due to the strong co-occurrence present in the group. In contrast, generative methods may benefit from the prevalence of common diseases during the SFT / RL learning phase. Notably, the distribution of generative methods is more uniform overall, especially in our approach. For instance, on the READ prediction task, our algorithm demonstrates superior performance on group G1 without being adversely affected by the more frequently occurring group G3. This uniformity can be attributed to two factors: first, it captures the semantic similarity between entities regardless of co-occurrence. This is crucial, as the semantics derived from the text are independent of the dataset's interactions, thus mitigating the effects of skewed distributions. Second, our model explores a wider range of potentially relevant entities during the deep thinking process, thus expanding the boundaries of knowledge. This capability enhances the effectiveness of our method across various disease categories. To sum up, adopting a fairer approach for the rare diseases group can lead to improved clinical outcomes by ensuring that patients receive more accurate diagnoses and tailored treatment options [33].

E. Semantic Understanding

In addition to evaluating healthcare generation, we conduct a thorough assessment of free-text generation performance, specifically focusing on question-answering (QA) tasks. To achieve this, we leverage the MMLU-Clinical [63] and MIMIC-IV-Ext-BHC datasets [64], where the input comprises rigorous clinical questions or original clinical notes. The objective is to generate multiple-choice answers or concise summaries formulated by expert clinicians. This task serves as a robust measure of the model's ability to produce coherent outputs and its depth of understanding in complex medical contexts. As demonstrated in Table VIII, our performance metrics in the medical QA tasks, including accuracy (ACC) and F1-score, indicate optimal results. Moreover, in the summary task, our ROUGE and SARI scores [65], [66] for this demanding free-text generation task are significantly elevated. In contrast, the MedRAG and KARE frameworks, which utilize the single-round retrieval approach, encounter substantial challenges in adapting to the deeper knowledge requirements inherent in these complex semantic understanding scenarios. This further highlights the superiority of our agentic RAG strategy, which proactively engages with semantic intricacies

to ensure precise comprehension of questions and optimal external knowledge integration.

TABLE VIII
SEMANTIC UNDERSTANDING. COMPARED TO TRADITIONAL HEALTHCARE PREDICTION TASKS, THIS APPROACH NO LONGER FOCUSES ON UNDERSTANDING PATIENT STATES; INSTEAD, IT EMPHASIZES UNDERSTANDING THE DIVERSE MEDICAL PROBLEMS AND THE MORE CHALLENGING TASK OF FREE-TEXT GENERATION.

Methods	QA		Summary		
	ACC↑	F1-score↑	ROUGE-1↑	ROUGE-L↑	SARI↑
DeepSeekR1-7B	0.3952	0.3921	0.2193	0.1188	0.3839
Biomistral-7B	0.2610	0.3572	0.1720	0.1016	0.3841
LightRAG	0.6913	0.6915	0.2719	0.1557	0.3995
Search-R1	0.7031	0.7039	0.2896	0.1611	0.3937
MedRAG	0.6991	0.7003	0.2642	0.1500	0.3990
KARE	0.7011	0.7015	0.2966	0.1758	0.4228
Medical-SFT	0.5992	0.6058	0.2458	0.1399	0.3859
Ours	**0.7167**	**0.7182**	**0.3090**	**0.1808**	**0.4307**

F. Complexity and Distributed Optimization

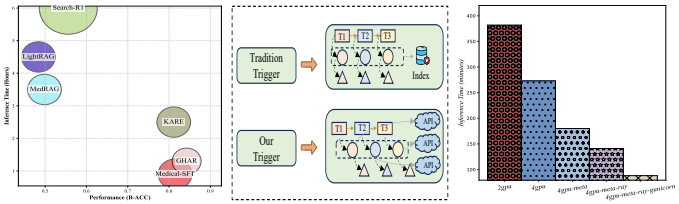
As shown in Fig. 8(a), we want to emphasize that under the same generative RAG paradigm, our method is more efficient. Operating directly on raw documents rather than on refined KG entities and relations, Search-R1 requires a more extensive exploration process and tokens, making it more susceptible to the "loss-in-the-middle" phenomenon. MedRAG and KARE perform retrieval indiscriminately across all queries, whereas our Agent-Top adaptively determines whether to invoke RAG based on specific needs, significantly reducing the costs associated with vector matching. Unlike KARE, we do not require synchronizing the output of the CoT at the training stage, which decreases the number of decoding operations needed—a process that can be very time-consuming when performed serially. Additionally, our two agents can essentially utilize the same LLM for optimization without introducing extra computational burden, maintaining a complexity level consistent with that of others during the inference phase.

Moreover, in our scenario, the input of a patient's complete longitudinal record within the same batch can be highly imbalanced, and the samples requiring the retrieval process are time-consuming. Therefore, performing LLM generation and retrieval sequentially for a batch is not advisable, as it results in slower speeds. This issue is not exclusive to our method but is common across most RAG-based approaches. To mitigate this, we optimize the overall computational framework through distributed strategies. We employ Ray for distributed request handling and utilize gunicorn to implement an asynchronous request system, as depicted in Fig. 8(b). Meanwhile, our retrieval process operates only on selected meta-path partitions, thereby reducing both the scope of indexing and the computational time required. As demonstrated in Fig. 8(c), these three strategies significantly reduce inference time, thereby enhancing the industrial applicability of our approach.

G. Case Studies

We conduct case studies to illustrate our rationale.

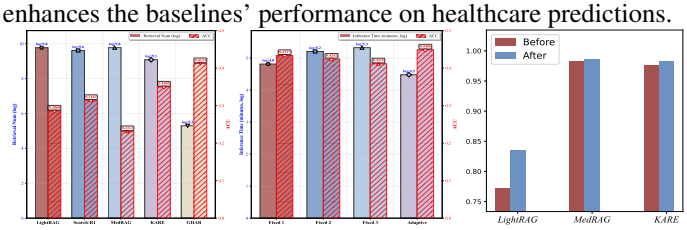
Iterative Deep Retrieval. We present the additional thinking processes of competitive RAG baselines and our algorithm in Fig. 9(a), which include the invocation of RAG operations and LLM-based reasoning. The results clearly demonstrate that our



(a) Complexity (b) Distributed Design (c) Time Reduction

Fig. 8. Time Complexity. To demonstrate practicality and fairness, for Fig. 8(a) and 8(c), we test the inference time for 1 epoch (MIMIC-III, DEC Pred) on a machine equipped with four RTX 3090 GPUs.

algorithm achieves optimal performance with the minimum number of additional thinking processes. To investigate the advantages of adaptive iteration, we replace the original adaptive decision-making mechanism of Agent-Top with a strategy that employs a fixed number of RAG calls. The results in Fig. 9(b) show a decline in performance, underscoring the importance of adaptively determining the depth of reasoning. A fixed number of iterations may lead to either overthinking or underthinking, consequently reducing accuracy for some questions. Additionally, our findings indicate that GHAR effectively identifies retrieval needs, mitigating potential issues of knowledge gaps or redundancy. This is further confirmed in Fig. 9(c), where eliminating unnecessary RAG pipelines enhances the baselines' performance on healthcare predictions.



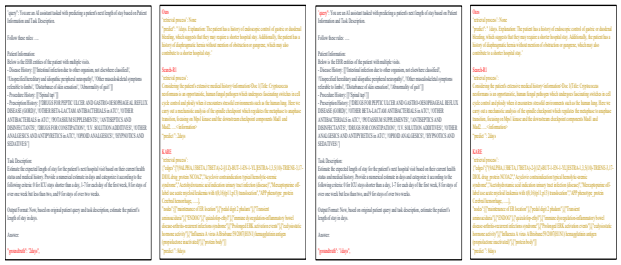
(a) Extra Thinking (b) Increased Iterations (c) RAG Pruning

Fig. 9. Iterative Deep Retrieval. (a) Extra Thinking & Performance on LOS Pred (MIMIC-III). (b) Examine the performance variation brought by growth iterations. (c) During the SFT phase of these algorithms, we remove the cases identified by our method as not requiring the RAG process and replace them with corresponding query-answer pairs as training data.

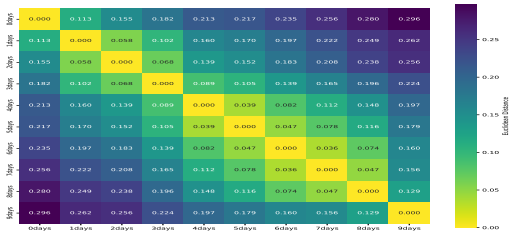
Illustrative Cases. In Fig. 10(a), we compare the knowledge extraction approaches of different algorithms. Our algorithm can correctly answer questions without the need for explicit knowledge extraction. In contrast, both Search-R1 and KARE encounter coarse-grained knowledge, which may lead the model to fall into a loss-in-middle scenario, resulting in incorrect answers. Another case study in Fig. 10(b) illustrates our algorithm's RAG-based extraction process. Our method conducts a two-iteration process in which the RAG component accesses only the partitions corresponding to the meta-path "('drug', 'drug_protein', 'gene/protein')", significantly reducing indexing time. Furthermore, from the perspective of sub-query reasoning, our algorithm iteratively identifies and addresses existing knowledge gaps (i.e., the second subquery), thereby enhancing overall performance.

Additionally, we emphasize that generative algorithms offer greater explainability compared to traditional discriminative algorithms. They not only provide rationales but also effectively capture the underlying correlations among different labels. This is convincingly demonstrated in Fig. 10(c),

which illustrates the representational similarities of responses across various labels. Specifically, the representation distance between responses for adjacent days is smaller, indicating a higher degree of similarity. This approach is cognitively aligned, as it captures the semantic relationships between labels. In contrast, discriminative methods treat labels as discrete, independent entities, merely predicting the most probable one without comprehending their underlying similarities.



(a) Inference Comparison (b) Our RAG Pipeline

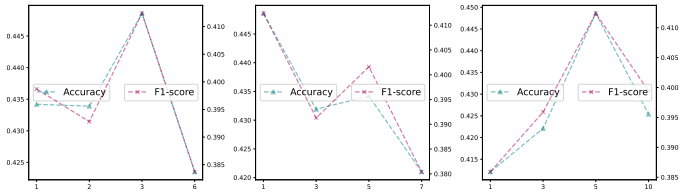


(c) Explainable Response

Fig. 10. Illustrative Cases. (a) Inference process for final prediction. (b) Another case of our retrieval process. (c) Explainable label relationships (MIMIC-III, LOS Pred).

H. Hyper-parameter Tests

We examine the essential hyperparameters in our experiments to achieve optimal performance.



(a) Meta-path Size (b) Top-K Recall (c) ORM Weight

Fig. 11. Hyper-parameter Tests. Here, we take the LOS Pred task (MIMIC-III) as an example.

Maximum Number of Meta-paths $|\tilde{\mathcal{O}}_c|$. It represents the breadth of meta-path selection. For complex issues, insufficient breadth may hinder thorough problem decomposition, potentially leading to unreasonable results. Conversely, excessive selection can introduce redundancy or irrelevant knowledge. Our experiments in Fig. 11(a) indicate that optimal performance is achieved when $|\tilde{\mathcal{O}}_c| = 3$. This suggests a balanced approach to the quantity of meta-paths, enabling effective problem-solving without unnecessary complexity.

Top-K Recall N. It indicates the number of nodes and edges matched during the RAG, which is crucial in GraphRAG-related work. A higher quantity provides a broader knowledge base, offering sufficient resources for Agent-Low to generate summaries. However, excessively high values may introduce irrelevant nodes and relationships from the KG, effectively adding noise and impacting subsequent knowledge refinement. According to Fig. 11(b), performance improves when $N = 1$.

ORM Reward Weight η. The ORM reward is a result-based incentive that serves as a shared objective for both models, in contrast to a standalone cost. A larger reward encourages collaboration between the two agents, while a smaller reward may lead the models to focus on their individual roles. As shown in Fig. 11(c), optimal performance is achieved when $\eta = 5$. Larger values may cause the sub-agents to neglect their own penalties, negatively impacting the stability of the reinforcement learning process as a whole.

VI. Conclusion

In this paper, we introduce GHAR, a framework that leverages hierarchical agentic RAG to enhance the performance of generative healthcare prediction. Our innovative design incorporates two distinct agents, Agent-Top and Agent-Low, which collaboratively determine when to engage in RAG and what information to retrieve. This iterative retrieval process explicitly integrates a reasoning chain into the model's decision-making framework, fostering more informed predictions. We also apply principles of multi-agent reinforcement learning, reformulating our approach as a MDP. Through a diverse reward structure, we achieve unified optimization of both the retrieval and generation modules, ensuring their seamless collaboration. Extensive experiments and robust analyses demonstrate the strong performance and interpretability of our model across various scenarios. Despite these advancements, our model has certain limitations. A key area for future work lies in designing more intuitive process rewards that could further enhance model performance.

VII. Prompt Templates

We provide the corresponding prompt templates.

Query Construction f_{query}

You are an AI assistant tasked with predicting a patient's next {task_info} based on Patient Information and Task Description.

Follow these rules:
1. Your prediction should be evidence-based and clinically relevant.
2. All predictions must trace to concrete EHR entities.

Patient Information: Below is the EHR entities of the patient with multiple visits.
- Disease History: {disease_info}
- Procedure History: {procedure_info}
- Prescription History: {medication_info}

Task Description: {task_desc}

Now, based on original patient query and task description, {question_answer_format}.

Answer:

Subquery Generation f_{generate}

You are an expert in query decomposition. Your task is to break down the following healthcare-related query into 3-5 clear and distinct subqueries.

Follow these rules:
1. Each subquery should focus on a specific aspect of the original query. Ensure logical independence.
2. Output Format: List of subqueries.

Examples:
Query: What are the health benefits of regular check-ups?

Subqueries:
- What are the benefits of regular health screenings?
- How do check-ups contribute to early disease detection?
- What is the recommended frequency of health check-ups for adults?

Now, decompose the following query:
Query: {query}
Subqueries:

Fig. 12. Prompt template for section III-C. The areas highlighted in red indicate the variables that are passed in. More task information and answer format can be found in Fig. 16.

RAG Decision $a_t^{T,1}$

You are an expert in query analysis. Your task is to decide whether the following healthcare subquery requires external knowledge retrieval or can be answered directly by an LLM.

Follow these rules:
1. If the subquery can be answered with general knowledge or simple reasoning by yourself, respond with 'no'. If the subquery requires specific medical knowledge, external data, or complex reasoning, respond with 'yes'.
2. Provide a confidence score (0 to 1) for your decision, where 1 means highly confident and 0 means not confident at all.
3. Use the reason history to support your answer with evidence.
4. Output Format: yes / no

Example:
Subquery: What are the common side effects of aspirin?
Reason History: Aspirin is widely used for pain relief and has known side effects.
Decision: no, Confidence Score: 0.9

Now, analyze the following healthcare subquery:
Subquery: {subquery}
Reason History: {reason_history}
Decision:

Terminal Decision $a_t^{T,2}$

You are an expert in evaluating reasoning completeness. Your task is to determine whether the provided reason history fully addresses the healthcare query.

Follow these rules:
1. Assess the query and the follow-up subquery-answer carefully.
2. If the reason history fully answers the query, respond with the final answer.
3. If it is incomplete or uncertain, respond with 'incomplete.'

Example:

Query: …. Will this patient be readmitted to the hospital within 15 days?
Reason History: ….. (too long, historical QA pair)
Answer: incomplete.

Example:
Query: ……How long will he live in the hospital?
Reason History: ….. (too long, historical QA pair)
Answer: 5 days

Now, evaluate the following:
Query: {query}
Reason History: {reason_history}
Answer:

Fig. 13. Prompt template for Agent-Top decision in section III-D.

LLM Generation f_{llm}

You are an expert in answering healthcare queries. Your task is to generate a final answer based on the user query and the provided reason history.

Follow these rules:
1.Ensure the answer directly addresses the user query.
2.Use the reason history to support your answer with evidence.
3.Output Format: Please be short and coherent.

Example:
Query: What are the benefits of regular exercise for heart health?
Reason History: Regular exercise can improve cardiovascular fitness, lower blood pressure, and reduce cholesterol levels.
Answer: Regular exercise significantly benefits heart health by improving cardiovascular fitness, lowering blood pressure, and reducing cholesterol levels.

Now, answer the following query based on the reason history:

Query: {subquery}
Reason History: {reason_history}
Answer:

Meta-paths Selection f_{meta}

You are an expert in meta-path analysis. Your task is to select the most relevant meta-paths for a given healthcare subquery based on the provided meta-path descriptions and reason history.

Follow these rules:
1. Analyze the subquery and the provided meta-path descriptions carefully.
2. Use the reason history to support your selection with evidence.
3. Output Format: List of ID numbers.

Example:
Subquery: What are the effects of diabetes on cardiovascular health?
Reason History: Previous studies indicate a strong correlation between diabetes and heart disease.
Meta-path Descriptions:
Meta-path 1: Diabetes → Cardiovascular Disease
Meta-path 2: Diabetes → Risk Factors → Cardiovascular …(too long, all meta-paths)

Selected Meta-path IDs: [1, 2]

Now, select the most relevant meta-paths for the following healthcare subquery:
Subquery: {subquery}
Reason History: {reason_history}
Meta-path Descriptions: {meta_path}
Selected Meta-path IDs:

Fig. 14. Fine-grained prompt template for Agent-Top action in section III-D.

RAG Generation f_{rag}

You are an expert in answering healthcare queries. Your task is to generate a final answer based on the user query and the provided summary.

Follow these rules:
1. Ensure the answer directly addresses the user query.
2. Use retrieval methods to support your answer with evidence.
3. Output Format: Please be short and coherent.

Example:
Query: What are the benefits of a balanced diet for overall health?
Knowledge Graph Results:
Nodes: Balanced Diet, Nutrients….
Edges: (Balanced Diet → Provides → Nutrients)…
Subgraph: ….(too long, a lot of triples)
Naive RAG Results: A balanced diet can lower the risk of chronic diseases and enhance mental well-being.
Answer: A balanced diet is vital for overall health as it provides essential nutrients and supports immune function while helping maintain a healthy weight.

Now, answer the following query based on the retrieval:
Query: {subquery}
Knowledge Graph Results: {kg_results}
Naive RAG Results: {naive_results}
Answer:

Deep Think f_{sub}

You are an expert in query generation. Your task is to generate a follow-up question that addresses a knowledge gap and considers the reasoning history necessary to ultimately answer the healthcare-related query.

Follow these rules:
1. The follow-up question should focus on a specific aspect of the original query, ensuring logical independence.
2. Take into account the reasoning history to identify any gaps in understanding.
3. Output Format: Please be short and coherent.

Example:
Query: What are the long-term effects of hypertension on kidney function?
Reason History: Hypertension can damage blood vessels and affect kidney health, but specific effects are unclear.
Follow-up Question:
What specific complications arise from kidney damage due to long-term hypertension?

Now, generate a follow-up question for the following query:
Query: {subquery}
Reason History: {reason_history}
Follow-up Question:

Fig. 15. Prompt template for Agent-Low action in section III-E and deep thinking in section III-A.

REFERENCES

[1] Z. Zheng, C. Wang, T. Xu, D. Shen, P. Qin, X. Zhao, B. Huai, X. Wu, and E. Chen, "Interaction-aware drug package recommendation via policy gradient," *TOIS*, vol. 41, no. 1, pp. 1–32, 2023.

[2] Z. Zheng, C. Wang, T. Xu, D. Shen, P. Qin, B. Huai, T. Liu, and E. Chen, "Drug package recommendation via interaction-aware graph induction," in *Proceedings of the Web Conference 2021*, 2021, pp. 1284–1295.

[3] X. Liu, H. Liu, G. Yang, Z. Jiang *et al.*, "A generalist medical language model for disease diagnosis assistance," *Nature Medicine*, pp. 1–11, 2025.

Task Description

Task_Info: Decompensation Prediction
Task Description: Assess the patient's overall health condition, considering the patient's disease history, procedures history, and prescription history to determine the likelihood of decompensation within 24 hours.
Q&A Format: If you determine that the patient is at significant risk of decompensation, respond with 'yes'; otherwise, respond with 'no'."

Task_Info: Readmission Prediction
Task Description: Predict the likelihood of the patient being readmitted to the hospital within the next 15 days based on their current health status and medical history.
Q&A Format: If you predict that the patient is likely to be readmitted, respond with 'yes'; otherwise, respond with 'no'.

Task_Info: Length-of-Stay Prediction
Task Description: Estimate the expected length of stay for the patient's next hospital visit based on their current health status and medical history.
Q&A Format: Provide a numerical estimate in days and categorize it according to the following criteria: 0 for ICU stays shorter than a day, 1-7 for each day of the first week, 8 for stays of over one week but less than two, and 9 for stays of over two weeks.

Semantic-QA

You are an AI assistant tasked with {task_info} a clinical note based on the initial clinical note provided.

Follow these rules:
1. Select the correct options for the multiple-choice questions based on the clinical note.
2. Provide brief explanations for your selected options to clarify your reasoning.

Initial Clinical Note: {initial_query}

Task Description: {task_desc}

Output Format:
Now, based on the initial clinical note, {question_answer_format}

Selected Options:

Task_Info: Multiple Q&A Choice
Task Description: Answer the multiple-choice questions based on the query. Provide explanations for the selected options and ensure clarity in your reasoning.
Q&A Format: Choose a correct option from [A, B, C, D] for the multiple-choice questions based on the clinical note. Be clear and concise.

Semantic-Summary

You are an AI assistant tasked with {task_info} a clinical note based on the initial clinical note provided.

Follow these rules:
1. Your summary should be clear, concise, and capture all essential details.
2. Focus on key symptoms, diagnosis, treatment plan, and relevant medical history.

Initial Clinical Note: {initial_clinical_note}

Task Description: {task_desc}

Output Format:
Now, based on the initial clinical note, {question_answer_format}

Summary:

Task_Info: Note Summary
Task Description: Summarize the clinical note by extracting key information about the patient's symptoms, diagnosis, treatment plan, and any relevant medical history.
Q&A Format: Give the summary of the clinical note. be short and concise.

Fig. 16. Prompt template for task description & semantic understanding in section V.

[4] Z. Chen, Y. Liao, S. Jiang, P. Wang, Y. Guo, Y. Wang, and Y. Wang, "Towards omni-rag: Comprehensive retrieval-augmented generation for large language models in medical applications," *CoRR*, vol. abs/2501.02460, 2025.

[5] Z. Wang, Y. Zhu, H. Zhao, X. Zheng, D. Sui, T. Wang, W. Tang, Y. Wang, E. Harrison, C. Pan *et al.*, "Colacare: Enhancing electronic health record modeling through large language model-driven multi-agent collaboration," in *WWW*, 2025, pp. 2250–2261.

[6] J. Gao, C. Xiao, Y. Wang, W. Tang, L. M. Glass, and J. Sun, "Stagenet: Stage-aware neural networks for health risk prediction," in *WWW*. ACM / IW3C2, 2020, pp. 530–540.

[7] C. Zhang, X. Gao, L. Ma, Y. Wang, J. Wang, and W. Tang, "Grasp: generic framework for health status representation learning based on incorporating knowledge from similar patients," in *AAAI*, vol. 35, no. 1, 2021, pp. 715–723.

[8] P. Jiang, C. Xiao, M. Jiang, P. Bhatia, T. A. Kass-Hout, J. Sun, and J. Han, "Reasoning-enhanced healthcare predictions with knowledge graph community retrieval," *ICLR*, 2025.

[9] C. Zhao, H. Tang, H. Zhao, and X. Li, "Beyond sequential patterns: Rethinking healthcare predictions with contextual insights," *TOIS*, 2025.

[10] Y. Chen, L. Yan, W. Sun, X. Ma, Y. Zhang, S. Wang, D. Yin, Y. Yang, and J. Mao, "Improving retrieval-augmented generation through multi-agent reinforcement learning," *NeurIPS*, 2025.

[11] P. Jiang, C. Xiao, A. Cross, and J. Sun, "Graphcare: Enhancing healthcare predictions with personalized knowledge graphs," in *ICLR*, 2024.

[12] Y. Zhu, C. Ren, Z. Wang, X. Zheng, S. Xie, J. Feng, X. Zhu, Z. Li, L. Ma, and C. Pan, "Emerge: Enhancing multimodal electronic health records predictive modeling with retrieval-augmented generation," in *CIKM*, 2024, pp. 3549–3559.

[13] Y. Xu, X. Jiang *et al.*, "Dearllm: Enhancing personalized healthcare via large language models-deduced feature correlations," in *AAAI*, vol. 39, no. 1, 2025, pp. 941–949.

[14] A. Bosselut, Z. Chen *et al.*, "Meditron: Open medical foundation models adapted for clinical practice," 2024.

[15] W. Nazar, G. Nazar *et al.*, "How to design, create, and evaluate an instruction-tuning dataset for large language model training in health care: Tutorial from a clinical perspective," *Journal of Medical Internet Research*, vol. 27, p. e70481, 2025.

[16] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *TOIS*, vol. 43, no. 2, pp. 1–55, 2025.

[17] Z. Wang, S. X. Teo, J. Ouyang, Y. Xu, and W. Shi, "M-RAG: reinforcing large language model performance through retrieval-augmented generation with multiple partitions," in *ACL*. Association for Computational Linguistics, 2024, pp. 1966–1978.

[18] Y. Li, X. Zhang, L. Luo, H. Chang, Y. Ren, I. King, and J. Li, "G-refer: Graph retrieval-augmented large language model for explainable recommendation," in *WWW*, 2025, pp. 240–251.

[19] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "Lightrag: Simple and fast retrieval-augmented generation," *CoRR*, vol. abs/2410.05779, 2024.

[20] X. Zhao, S. Liu, S.-Y. Yang, and C. Miao, "Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot," in *WWW*, 2025, pp. 4442–4457.

[21] B. Jin, H. Zeng, Z. Yue, D. Wang, H. Zamani, and J. Han, "Search-r1: Training llms to reason and leverage search engines with reinforcement learning," *CoRR*, vol. abs/2503.09516, 2025.

[22] H. Lee, L. Soldaini, A. Cohan, M. Seo, and K. Lo, "Routerretriever: Routing over a mixture of expert embedding models," in *AAAI*, vol. 39, no. 11, 2025, pp. 11 995–12 003.

[23] J. Yu, Q. Ge, X. Li, and A. Zhou, "Heterogeneous graph contrastive learning with meta-path contexts and adaptively weighted negative samples," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 10, pp. 5181–5193, 2024.

[24] X. Wang, D. Bo *et al.*, "A survey on heterogeneous graph embedding: methods, techniques, applications and sources," *IEEE Transactions on Big Data*, vol. 9, no. 2, pp. 415–436, 2022.

[25] M. Lu, Y. Min, Z. Wang, and Z. Yang, "Pessimism in the face of confounders: Provably efficient offline reinforcement learning in partially observable markov decision processes," in *ICLR*, 2023.

[26] W.-K. Ching and M. K. Ng, "Markov chains," *Models, algorithms and applications*, vol. 650, pp. 111–139, 2006.

[27] K. Hu, M. Li, Z. Song, K. Xu, Q. Xia, N. Sun, P. Zhou, and M. Xia, "A review of research on reinforcement learning algorithms for multi-agents," *Neurocomputing*, p. 128068, 2024.

[28] B. Jiang, Y. Xie, X. Wang, W. J. Su, C. J. Taylor, and T. Mallick, "Multi-modal and multi-agent systems meet rationality: A survey," in *ICML 2024 Workshop on LLMs and Cognition*, 2024.

[29] Q. Liu, X. Wu, X. Zhao, Y. Zhu, D. Xu, F. Tian, and Y. Zheng, "When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications," in *SIGIR*, 2024, pp. 1104–1114.

[30] C. Chen, J. Yu, S. Chen, C. Liu, Z. Wan, D. S. Bitterman, F. Wang, and K. Shu, "Clinicalbench: Can llms beat traditional ML models in clinical prediction?" *NeurIPS*, 2025.

[31] C. Zhao, H. Zhao, X. Zhou, and X. Li, "Enhancing precision drug recommendations via in-depth exploration of motif relationships," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8164–8178, 2024.

[32] S. Liu, X. Wang, J. Du, Y. Hou, X. Zhao, H. Xu, H. Wang, Y. Xiang, and B. Tang, "Shape: A sample-adaptive hierarchical prediction network for medication recommendation," *IEEE Journal of Biomedical and Health Informatics*, 2023.

[33] C. Zhao, H. Tang, J. Zhang, and X. Li, "Unveiling discrete clues: Superior healthcare predictions for rare diseases," in *WWWW*, 2025, pp. 1747–1758.

[34] J. Liu, Z. Huang, Q. Liu, Z. Ma, C. Zhai, and E. Chen, "Knowledge-centered dual-process reasoning for math word problems with large language models," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[35] Z. Chen, A. Hernández-Cano, A. Romanou, A. Bonnet, K. Matoba, F. Salvi, M. Pagliardini, S. Fan, A. Köpf, A. Mohtashami, A. Sallinen, A. Sakhaeirad, V. Swamy, I. Krawczuk, D. Bayazit, A. Marmet, S. Montariol, M. Hartley, M. Jaggi, and A. Bosselut, "MEDITRON-70B: scaling medical pretraining for large language models," *CoRR*, vol. abs/2311.16079, 2023.

[36] Y. Labrak, A. Bazoge, E. Morin, P. Gourraud, M. Rouvier, and R. Dufour, "Biomistral: A collection of open-source pretrained large language models for medical domains," in *ACL*. Association for Computational Linguistics, 2024, pp. 5848–5864.

[37] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A survey on rag meeting llms: Towards retrieval-augmented large language models," in *SIGKDD*, 2024, pp. 6491–6501.

[38] X. Guan, J. Zeng, F. Meng, C. Xin, Y. Lu, H. Lin, X. Han, L. Sun, and J. Zhou, "Deeprag: Thinking to retrieval step by step for large language models," *CoRR*, vol. abs/2502.01142, 2025.

[39] L. M. Amugongo, P. Mascheroni, S. Brooks, S. Doering, and J. Seidel, "Retrieval augmented generation for large language models in healthcare: A systematic review," *PLOS Digital Health*, vol. 4, no. 6, p. e0000877, 2025.

[40] O. Huly, I. Pogrebinsky, D. Carmel, O. Kurland, and Y. Maarek, "Old ir methods meet rag," in *SIGIR*, 2024, pp. 2559–2563.

[41] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering." in *EMNLP*, 2020, pp. 6769–6781.

[42] H. Yu, J. Kang, R. Li, Q. Liu, L. He, Z. Huang, S. Shen, and J. Lu, "Cagar: Context-aware alignment of llm generation for document retrieval," in *ACL*, 2025, pp. 5836–5849.

[43] Y. Gao, Y. Xiong, M. Wang, and H. Wang, "Modular RAG: transforming RAG systems into lego-like reconfigurable frameworks," *CoRR*, vol. abs/2407.21059, 2024.

[44] Y. Li, P. Wang, X. Zhu, A. Chen, H. Jiang, D. Cai, V. W. Chan, and J. Li, "Glbench: A comprehensive benchmark for graph with large language models," *NeurIPS*, vol. 37, pp. 42 349–42 368, 2024.

[45] L. Wu, Z. Li, H. Zhao, Z. Huang, Y. Han, J. Jiang, and E. Chen, "Supporting your idea reasonably: A knowledge-aware topic reasoning strategy for citation recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 8, pp. 4275–4289, 2024.

[46] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, "Agentic retrieval-augmented generation: A survey on agentic RAG," *CoRR*, vol. abs/2501.09136, 2025.

[47] Q. Team, "Qwen2.5: A party of foundation models," September 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen2.5/

[48] P. Chandak, K. Huang, and M. Zitnik, "Building a knowledge graph to enable precision medicine," *Scientific Data*, vol. 10, no. 1, p. 67, 2023.

[49] DeepSeek-AI, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025.

[50] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" in *EMNLP*. Association for Computational Linguistics, 2022, pp. 11 048–11 064.

[51] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, vol. 33, pp. 1877–1901, 2020.

[52] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *NeurIPS*, vol. 35, pp. 24 611–24 624, 2022.

[53] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual E5 text embeddings: A technical report," *CoRR*, vol. abs/2402.05672, 2024.

[54] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for page-wise recommendations," in *Proceedings of the 12th ACM conference on recommender systems*, 2018, pp. 95–103.

[55] C. Yang, Z. Wu *et al.*, "Pyhealth: A deep learning toolkit for healthcare applications," in *SIGKDD*. ACM, 2023, pp. 5788–5789.

[56] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[57] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow *et al.*, "Mimic-iv, a freely accessible electronic health record dataset," *Scientific data*, vol. 10, no. 1, p. 1, 2023.

[58] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eicu collaborative research database, a freely available multi-center database for critical care research," *Scientific data*, vol. 5, no. 1, pp. 1–13, 2018.

[59] M. Xu, Z. Zhu, Y. Li, S. Zheng, Y. Zhao, K. He, and Y. Zhao, "Flexcare: Leveraging cross-task synergy for flexible multimodal healthcare prediction," in *SIGKDD*, 2024, pp. 3610–3620.

[60] Y. Zhong, S. Cui, J. Wang, X. Wang, Z. Yin, Y. Wang, H. Xiao, M. Huai, T. Wang, and F. Ma, "Meddiffusion: Boosting health risk prediction via diffusion-based data augmentation," in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 2024, pp. 499–507.

[61] J. Chen, S. Xiao *et al.*, "BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *CoRR*, vol. abs/2402.03216, 2024.

[62] G. Wang, X. Liu, Z. Ying, G. Yang, Z. Chen, Z. Liu, M. Zhang, H. Yan, Y. Lu, Y. Gao *et al.*, "Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial," *Nature Medicine*, vol. 29, no. 10, pp. 2633–2642, 2023.

[63] S. Kim and H.-J. Yoon, "Questioning our questions: How well do medical qa benchmarks evaluate clinical capabilities of language models?" in *Proceedings of the 24th Workshop on Biomedical Language Processing*, 2025, pp. 274–296.

[64] A. Aali, D. Van Veen, Y. Arefeen, J. Hom, C. Bluethgen, E. P. Reis, S. Gatidis, N. Clifford, J. Daws, A. Tehrani *et al.*, "Mimic-iv-ext-bhc: labeled clinical notes dataset for hospital course summarization," *PhysioNet*, 2024.

[65] L. Ermakova, J. V. Cossu, and J. Mothe, "A survey on evaluation of summarization methods," *Information processing & management*, vol. 56, no. 5, pp. 1794–1814, 2019.

[66] Y. Zhu, Z. He, H. Hu *et al.*, "Medagentboard: Benchmarking multi-agent collaboration with conventional methods for diverse medical tasks," *NeurIPS*, 2025.