Cite What You Explore: Budget-Aware LLM Reasoning over Medical KGs with Verifiable Evidence

Chen Chen[†], **Dongjie Wang**[†], **Mei Liu**[‡], **Zijun Yao**[†*]

[†]Electrical Engineering and Computer Science, University of Kansas, USA
{chenchen, wangdongjie, zyao}@ku.edu

[‡]Health Outcomes and Biomedical Informatics, University of Florida, USA
mei.liu@ufl.edu

Abstract

Post-discharge risk prediction from electronic health records (EHRs) is difficult because many dependencies that link discharge-time observations to downstream complications, such as comorbidity cascades and drug-disease interactions, are absent from the record. External medical knowledge graphs (KGs) can supply these missing dependencies, but tracing them in a clinical setting demands three properties. KG exploration must remain cost-bounded, retrieved evidence must be differentiated by source quality, and the resulting rationale must be citable for review. Large language models (LLMs) can plan and verify over structured evidence, making them natural candidates for KG reasoning, but existing LLM-based methods do not satisfy these three properties jointly. In this paper, we propose **BAR**, a **B**udget-**A**ware LLM **R**easoning framework over medical KGs with three contributions. First, BAR refines the raw KG into disease-specific evidence graphs whose edges carry support scores and provenance records, turning the KG into a quality-annotated reasoning space rather than a static feature source. Second, an LLM reasons over this graph through a plan-navigate-verify loop that decomposes the question into steps, retrieves evidence under a patient-specific budget, and revises when verification fails. Third, a reasoning policy is trained with a reward that compares predictions with and without acquired evidence, combined with acquisition cost and citation-integrity terms. Across 8 diseases and 3 prediction horizons on MIMIC-III and MIMIC-IV, BAR achieves higher performance and citation precision than strong baselines while using fewer LLM tokens. Its gain over EHR-only prediction grows at longer horizons, and its advantage carries over to an external benchmark.

1 Introduction

Electronic health records (EHRs) have enabled a wide range of post-discharge risk prediction tasks [31, 15], where the goal is to assess at discharge whether a specific complication will first appear within a future time window. This task is challenging because the pathways that would connect discharge-time observations to downstream complications, including comorbidity cascades, drug-disease interactions, and disease progressions, are rarely captured in the record. External medical knowledge graphs (KGs) [5] can supply the relational structure needed to recover these dependencies, but tracing them typically requires multi-hop exploration [41], since a single retrieval rarely captures the chain of intermediate concepts linking a discharge-time observation to its downstream complication.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing methods do not yet exploit this structure effectively. EHR-only models [9, 13, 32] capture statistical associations but miss risk factors that emerge only through pathways whose intermediate concepts the EHR does not record. Static KG methods [8, 26, 18, 34] add relational knowledge but extract embeddings once before prediction, with no mechanism to follow patient-specific pathways at inference time. What is missing is an agent that can decide which pathway to trace next and when the evidence is sufficient to stop for a given patient. Large language models (LLMs) are a natural fit [33, 23] because they can decompose a clinical scenario into hypothesis-driven steps and query the KG for evidence at each step, conditioning each retrieval on what previous steps have revealed.

Recent Graph4LLM approaches [42, 30, 12, 6] advance in this direction by allowing iterative reasoning over KGs, and medical KG methods such as KARE [19] retrieve summarized graph communities for each patient. However, three requirements for clinical use remain unmet. First, with no cost constraint, exploration continues well past the point of diminishing returns, even on patients whose discharge record alone already supports a confident prediction. Second, treating all retrieved edges uniformly, regardless of how well-supported they are by underlying evidence, leaves the LLM unable to prioritize the most reliable pathways. Third, the reasoning trace does not preserve a verifiable link between predictions and the provenance of supporting edges, so it cannot be audited.

To address these three requirements, we give the LLM and the KG complementary roles. The LLM generates and revises pathway hypotheses, while the KG constrains which pathways are structurally grounded and supplies the traceable evidence needed to verify them under bounded cost. We develop this idea in BAR[2] (Figure 1), a framework for budget-aware LLM reasoning over medical KGs. Our contributions are threefold. First, we introduce evidence graphs that refine the raw KG into disease-specific subgraphs whose edges carry support scores, provenance records, and equivalence-group ids that control redundancy, turning the KG into a quality-annotated reasoning space the LLM explores at inference time, rather than a feature source whose embeddings are extracted once before prediction. Second, we propose a budget-aware reasoning loop that allocates exploration according to the training-set percentile of EHR-only prediction uncertainty, in which the LLM produces a clinical plan, navigates the evidence graph under this patient-adaptive budget, and verifies each step against both source support and the patient's evolving state, with revision when evidence falls short. Third, we design a training signal whose primary term is a paired gain comparing predictions with and without acquired evidence, with auxiliary terms for acquisition cost and citation integrity, so the policy learns to acquire external evidence only when it improves prediction beyond what the discharge record alone provides. Experiments on MIMIC-III [22] and MIMIC-IV [21] show that BAR achieves higher AUROC and AUPRC than the compared baselines with fewer acquisition attempts and LLM tokens, and that a larger share of its citations is judged relevant. Its relative gain over EHR-only prediction increases at longer horizons, and its advantage carries over to the external EHRSHOT benchmark [37].

2 Methodology

2.1 Problem Setup

We predict the first recorded diagnosis of a target disease during a subsequent admission in the available EHR. Each eligible admission defines an instance for each target disease for which it meets the eligibility criteria, and discharge is the prediction time. The index i denotes an index admission and its associated patient history, with discharge time τ_i. The model input available at discharge has two parts, the diagnosis codes recorded during the index admission, $X_i = \{c_{i,1}, \ldots, c_{i,n_i}\}$, where each $c_{i,j}$ is an ICD code, and a pre-discharge history H_i that aggregates diagnosis codes from admissions strictly before τ_i.

Let $\mathcal{D}$ denote a predefined set of target diseases and $\mathcal{T}$ denote a set of prediction horizons in days. For index admission i, target disease $d \in \mathcal{D}$, and horizon $h \in \mathcal{T}$, the binary label is

$$y_{i,d}^{(h)} = \mathbb{1}[\exists\, t \in (\tau_i,\, \tau_i + h] \text{ such that } d \text{ is first recorded at time } t]\,. \tag{1}$$

To focus on the first recorded post-discharge occurrence, we restrict to instances where d has not appeared in either H_i or X_i, and require a positive outcome to be documented during a subsequent admission. First occurrence is defined from the available recorded history. Appendix A specifies

[2]Code is available at `https://github.com/ChenC2002/BAR`.

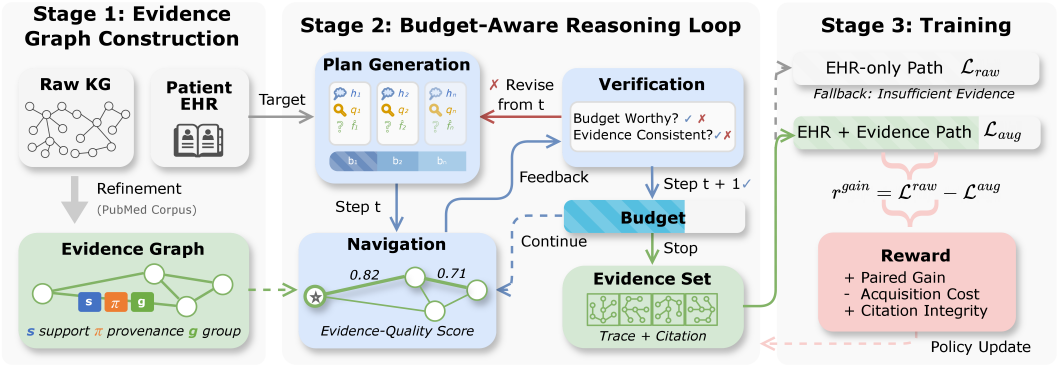

Figure 1: Overview of BAR. **Stage 1:** The raw KG is refined into a disease-specific evidence graph whose edges carry support scored against curated sources and a fixed PubMed corpus, provenance, and deduplication metadata. **Stage 2:** The LLM generates a budget-aware reasoning plan, navigates the evidence graph using a patient-conditioned quality score, and verifies each step for consistency and the cost-quality trade-off, revising as needed. **Stage 3:** The reasoning policy is trained via paired gain, acquisition cost, and citation-integrity reward.

the outcome timestamp and reports admission-level counts. A zero label indicates no qualifying diagnosis within the prediction window, including instances with no recorded subsequent admission. A patient may contribute multiple eligible index admissions, and one subsequent disease record can yield positive labels for multiple earlier eligible indices whose prediction windows include it. The model outputs a risk score $\hat{y}_{i,d}^{(h)} \in [0,1]$ together with a cited evidence trace whose supporting edges can be inspected through stored provenance.

We represent an external biomedical knowledge graph such as PrimeKG [5] as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For each target disease d, we construct a refined evidence graph $G_d = (V_d, E_d)$ from $\mathcal{G}$. Each edge $e = (u, r, v) \in E_d$ is a typed triple with endpoint concepts $u, v \in V_d$ and relation type r. Every retained edge carries per-edge metadata defined during evidence graph construction (Section 2.2.2).

We encode the patient's discharge-time EHR into a representation $\mathbf{z}_i^{\text{ehr}}$ that summarizes X_i and H_i. To connect discharge-time observations to the evidence graph, we use a fixed linker $\text{link}(\cdot)$ that maps ICD codes to concepts in $\mathcal{V}$. For patient i and target disease d, the anchor set is $A_{i,d} = \{\text{link}(c) : c \in X_i,\ \text{link}(c) \in V_d\}$. All graph interaction during reasoning is performed under a patient-specific budget B_i.

2.2 Refined Evidence Graph

This stage converts the raw knowledge graph into a disease-specific reasoning space that retains only supported, traceable, and low-redundancy edges. Each graph is built per disease and shared across horizons, which enter later through the target embedding and the prediction head (Sections 2.3.1 and 2.4.1), so different horizons can follow different paths through the same graph.

2.2.1 Candidate Region

Starting from disease node d, we collect all edges within K hops, where K is a fixed hop limit. Let $\mathcal{E}_{\leq K}(d)$ denote the set of edges in $\mathcal{E}$ whose endpoints are both within K hops of d. For reachability, we treat the graph as undirected, while each retained edge keeps its original relation type and direction.

We then restrict this region to edges connected to discharge-time observations. Let $\mathcal{A}_d$ denote the set of frequent anchor concepts for disease d, obtained by applying $\text{link}(\cdot)$ to frequent discharge-time codes in the training split. We keep only edges that lie on at least one path from an anchor to the disease node:

$$E_d^{\text{cand}} = \{e \in \mathcal{E}_{\leq K}(d) \mid \exists\, a \in \mathcal{A}_d \text{ such that } e \text{ lies on a path from } a \text{ to } d\}\,. \tag{2}$$

To reduce hub-driven noise, we apply a fixed degree threshold $d_{\max}$. For any node whose degree exceeds $d_{\max}$ in the candidate region, we keep at most M incident edges ranked by a fixed prior score $\tilde{s}(e) \in [0, 1]$ that is available before support scoring.

2.2.2 Support, Provenance, and Deduplication

For each candidate edge $e = (u, r, v) \in E_d^{\text{cand}}$, we retrieve supporting sources and store their identifiers as the provenance record $\pi(e)$. The sources are curated records from PrimeKG's constituent resources [5] and abstracts from a fixed PubMed corpus. The provenance record identifies the source and its release or publication date. Appendix B.1 specifies a fixed-snapshot construction and its temporal scope. A fixed support scorer assigns $s(e) \in [0, 1]$ using only the sources in $\pi(e)$. The score is 0.4 times a curated term, which counts the curated resources that attest the edge, plus 0.6 times a text term, which measures the PubMedBERT [14] similarity between the edge and its top three BM25-retrieved abstracts (Appendix B.1). We retain only edges with $s(e) \geq \tau_s$, yielding the filtered set $E_d^{\text{keep}} \subseteq E_d^{\text{cand}}$. With $\tau_s = 0.3$, an edge attested by three curated resources is retained without textual support, while an edge with fewer attestations needs some literature support.

To reduce redundancy, we group semantically equivalent edges in E_d^{keep} and assign each edge an equivalence-group id $g(e)$. Two edges are grouped only if they express the same medical fact with the same direction and comparable specificity. Each group keeps one representative edge for citation, while all member edges are retained for traceability.

We also store a lightweight expansion guide I_d for frequent anchors. For each anchor $a \in \mathcal{A}_d$, the guide keeps a short ranked list of adjacent edge identifiers in E_d. The final evidence graph is $G_d = (V_d, E_d)$, where $E_d = E_d^{\text{keep}}$.

2.3 Budget-Aware Reasoning Loop

Given a patient, a target disease, and a prediction horizon, this stage runs an LLM reasoning loop over G_d under budget B.

2.3.1 Evidence-Quality Score

We first define a patient-conditioned quality score used throughout the loop. Because the same discharge record may be relevant to different diseases and horizons in different ways, we condition the patient state on the prediction target. For each target pair (d, h), the initial patient state is $\mathbf{p}_{i,d,h} = f_{\text{state}}([\mathbf{z}_i^{\text{ehr}};\ \mathbf{u}(d, h)])$, where $f_{\text{state}}(\cdot)$ is a learnable projection, $\mathbf{u}(d, h) \in \mathbb{R}^m$ is a learnable target embedding, and $[\cdot\,;\,\cdot]$ denotes concatenation. After a step passes verification, the patient state updates via the GRU-based [7] transition $\mathbf{p}_{i,d,h}^{(t)} = \text{GRU}(\mathbf{p}_{i,d,h}^{(t-1)}, \bar{\mathbf{v}}^{(t)})$, where $\bar{\mathbf{v}}^{(t)}$ is the mean representation of edges selected at step t. Failed steps leave the patient state unchanged. All parameters of the reasoning policy are shared across diseases and horizons. Only the lightweight target embedding $\mathbf{u}(d, h)$ and the prediction heads are task-specific.

Each concept node $x \in V_d$ has a learnable embedding $\mathbf{e}_x \in \mathbb{R}^m$. For a candidate edge $e = (u, r, v) \in E_d$ under patient state $\mathbf{p}$, we define the evidence-quality score as

$$\phi(e;\, \mathbf{p}) = s(e)\ \sigma(\beta \cdot \max\{\mathbf{p} \cdot \mathbf{e}_u,\ \mathbf{p} \cdot \mathbf{e}_v\})\,, \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function and $\beta > 0$ is a fixed temperature that controls the sharpness of patient-relevance gating without adding a learnable scaling parameter. We write $\phi_t(e)$ for the score under the step-t state and $\phi^\star(e)$ for the score under the terminal state. The multiplicative form acts as a soft AND gate, requiring an edge to be both well-supported and patient-relevant to receive a high score. Appendix G analyzes the joint effect of support and patient relevance on selection rates.

2.3.2 Budget and Cost

When the discharge record already supports a confident prediction, additional evidence acquisition adds cost without improving accuracy, so we allocate larger budgets to uncertain EHR-only predictions from stage 1.

For target (d, h), let $\hat{y}_{i,d,h}^{\text{ehr}}$ be the frozen EHR-only score. We define $\tilde{u}_i = 1 - |2\hat{y}_{i,d,h}^{\text{ehr}} - 1|$ and use its training-set percentile $u_i = F_{d,h}^{\text{train}}(\tilde{u}_i)$. Uncertainty is highest near a predicted risk of 0.5. The CDF

$F^{\text{train}}_{d,h}$ is estimated separately for each disease and horizon and held fixed for validation and testing. We omit the fixed target from uncertainty and budget subscripts. The patient-specific budget is

$$B_i = B_{\min} + (B_{\max} - B_{\min}) \cdot u_i, \tag{4}$$

where $B_{\min} = 2$ allows an initial acquisition attempt and $B_{\max} = 12$ caps total exploration.

Within this budget, each acquisition step t incurs a cost determined by three binary indicators. The indicators δ^q_t and δ^{exp}_t record whether QUERYGUIDE or QUERYEXPAND is executed. These query types are mutually exclusive. The indicator δ^{sel}_t records whether the selection operation is executed. An executed selection operation is charged even when it returns no edges, and subsequent verification failure does not refund the cost. The per-step cost is

$$c_t = \alpha_q \delta^q_t + \alpha_{\text{exp}} \delta^{\text{exp}}_t + \alpha_{\text{sel}} \delta^{\text{sel}}_t, \tag{5}$$

where $\alpha_q, \alpha_{\text{exp}}, \alpha_{\text{sel}} \geq 0$ are fixed weights. The cumulative cost up to step t is $C_t = \sum_{j=1}^{t} c_j$.

2.3.3 Plan Generation

The LLM produces an initial reasoning plan that decomposes the clinical question into a sequence of evidence-gathering steps. Given patient EHR (X_i, H_i), target disease d, horizon h, and anchor set $A_{i,d}$, we prompt the LLM to produce a plan $P = [s_1, s_2, \ldots, s_n]$, where n is determined by the LLM. Each step $s_t = (h_t, q_t, \hat{f}_t, b_t)$ contains a clinical hypothesis h_t, a query intent q_t, an expected evidence description $\hat{f}_t$, and a planned budget fraction $b_t \in (0, 1]$, with $\sum_t b_t \leq 1$. The fractions guide the LLM's planning and revision. Execution enforces the episode-level budget by requiring each acquisition cost to fit within the remaining budget $B_i - C_{t-1}$.

2.3.4 Plan-Guided Graph Navigation

At plan step t, the query intent q_t guides retrieval from G_d through two query types. QUERYGUIDE(a) starts from an anchor $a \in A_{i,d}$ and returns candidates from the expansion guide I_d when $a \in \mathcal{A}_d$, falling back to direct lookup of incident edges otherwise. QUERYEXPAND(x, r) expands a frontier concept x along relation type r and returns incident edges in E_d. The first step of an episode uses QUERYGUIDE to ground the reasoning in the patient's discharge-time observations, while subsequent steps use QUERYEXPAND to follow the pathway outward from concepts discovered in earlier steps.

After a query returns candidates, we score each with $\phi_t(e)$, keep only the top K_{pre}, and remove any candidate whose equivalence-group id has already been selected. The filtered candidates are presented to the LLM as graph feedback f_t, consisting of typed triples with their support scores.

The LLM then selects edges to retain. Each selected edge must satisfy three executable constraints requiring that the edge exists in E_d, its support score satisfies $s(e) \geq \tau_s$, and it belongs to an equivalence group that has not been previously selected. Selected edges are added to the current evidence set $E^{(t)}_{i,d,h}$ only if the step passes verification.

Coherence comes from three inductive biases. The patient state aggregates previously selected evidence through recurrent updates, expansion starts from anchors or endpoints of accepted evidence, and equivalence-group exclusion prevents redundant relations.

2.3.5 Verification and Revision

After each navigation step, the LLM compares the graph feedback f_t against its expected evidence $\hat{f}_t$ and triggers a revision when the feedback contradicts or fails to support its hypothesis. Verification proceeds in two stages. First, deterministic admissibility checks enforce hard constraints on support, equivalence-group uniqueness, and minimum evidence quality. Once these checks pass, the LLM verifies semantically whether the evidence coherently supports the clinical hypothesis.

On failure, the evidence set and patient state remain unchanged. If enough budget remains, the LLM regenerates the remaining plan from the failed step onward, keeping verified earlier steps unchanged. The revision prompt includes the current feedback, the accumulated reasoning context, and the remaining budget $B_i - C_t$, allowing the LLM to adapt plan length to actual consumption.

2.3.6 Stopping

To detect when further exploration yields diminishing returns, we track the set-level utility

$$U_t = \frac{1}{\max\{1,\, |E^{(t)}_{i,d,h}|\}} \sum_{e \in E^{(t)}_{i,d,h}} \phi_t(e), \tag{6}$$

where all selected edges are rescored under the current patient state at step t.

The loop terminates when the remaining budget $B_i - C_t$ cannot support another feasible action, when U_t improves by less than ϵ_{stop} for P_{stop} consecutive steps, or when the evidence set reaches a size cap K_E. If the final evidence set is too small ($|E_{i,d,h}| < K_{\min}$) or too weak ($U_{\text{final}} < \tau_{\text{fb}}$), the model falls back to the EHR-only prediction.

2.4 Prediction and Training

2.4.1 Predictor

For episodes that pass the fallback thresholds, we summarize the final evidence set $E_{i,d,h}$ using score-weighted pooling. Let $\mathbf{v}(e) \in \mathbb{R}^p$ denote the learned representation of edge e. The evidence representation is

$$\mathbf{z}^{\text{kg}}_{i,d,h} = \sum_{e \in E_{i,d,h}} w(e)\, \mathbf{v}(e), \qquad w(e) = \frac{\phi^\star(e)}{\max\Big(\epsilon,\, \sum_{e' \in E_{i,d,h}} \phi^\star(e')\Big)}, \tag{7}$$

where $\epsilon > 0$ is a small constant. On fallback, the model returns the frozen EHR-only score $\hat{y}^{\text{ehr}}_{i,d,h}$ and an empty citation list. Otherwise, the predictor combines $\mathbf{z}^{\text{ehr}}_i$ from Section 2.1 with the evidence representation. The final risk score is

$$\hat{y}^{(h)}_{i,d} = \sigma\Big(\mathbf{t}^\top_{d,h} f\Big([\mathbf{z}^{\text{ehr}}_i;\, \mathbf{z}^{\text{kg}}_{i,d,h}]\Big) + b_{d,h}\Big), \tag{8}$$

where $f(\cdot)$ is a shared trainable fusion function with output in $\mathbb{R}^p$. Each disease-horizon pair has a prediction head with parameters $\mathbf{t}_{d,h} \in \mathbb{R}^p$ and $b_{d,h} \in \mathbb{R}$. The LLM therefore controls evidence acquisition, and its plans, hypotheses, budget fractions, and verification decisions influence the risk score only through the edges that enter $E_{i,d,h}$. The citation list is formed by sorting edges in $E_{i,d,h}$ by $\phi^\star(e)$, deduplicating by $g(e)$, and keeping at most K_C edges, each retaining its support score, provenance record, and equivalence-group membership for inspection.

2.4.2 Reward

We compare predictions with and without acquired evidence under the same model snapshot. Let $\mathcal{L}^{\text{raw}}_{i,d,h}$ denote the binary cross-entropy loss when the evidence representation is set to zero, and let $\mathcal{L}^{\text{aug}}_{i,d,h}$ denote the loss when $E_{i,d,h}$ is used. The paired gain is

$$r^{\text{gain}}_{i,d,h} = \mathcal{L}^{\text{raw}}_{i,d,h} - \mathcal{L}^{\text{aug}}_{i,d,h}. \tag{9}$$

Both losses are computed on the same batch with dropout disabled.

The citation-integrity reward $r^{\text{qual}}_{i,d,h}$ is 1 when the final citation list is non-empty and passes all integrity checks, and 0 otherwise. Every cited edge must belong to E_d, satisfy $s(e) \geq \tau_s$, and have a non-empty provenance record. The list must contain at most K_C edges with distinct equivalence-group identifiers. Fallback episodes receive zero because they return no citations. Patient-specific relevance is evaluated separately by the citation judge. The full episode reward is

$$r_{i,d,h} = \lambda_{\text{gain}}\, r^{\text{gain}}_{i,d,h} - \lambda_{\text{cost}}\, \frac{C_{i,d,h}}{B_{\max}} + \lambda_{\text{qual}}\, r^{\text{qual}}_{i,d,h}, \tag{10}$$

where $C_{i,d,h} = \sum_t c_t$ is the total acquisition cost, $B_{\max}$ is the maximum budget and serves as a fixed normalizer, and $\lambda_{\text{gain}}, \lambda_{\text{cost}}, \lambda_{\text{qual}} \geq 0$ are fixed weights.

2.4.3 Training Procedure

Training proceeds in three stages, detailed in Appendix B. In the first stage, we train an EHR-only baseline by setting $\mathbf{z}^{\mathrm{kg}}_{i,d,h} = \mathbf{0}$ for all instances and optimizing the predictor with binary cross-entropy loss. This baseline is frozen for budget allocation and fallback. The paired losses in Eq. (9) use the current predictor with and without acquired evidence.

In the second stage, the reasoning loop runs with a fixed exploration policy that executes a small number of queries and selections per episode, without policy optimization. The predictor is updated with supervised loss while conditioning on the acquired evidence.

In the third stage, we update the reasoning policy using REINFORCE [40] with the episode reward in Eq. (10), while continuing supervised predictor updates. Appendix B.3 specifies the routing distribution, trainable parameters, and update, and Appendix B reports the optimization settings.

3 Experiments

3.1 Setup

Datasets. We evaluate on MIMIC-III [22, 15] and MIMIC-IV [21]. We define 8 target diseases spanning comorbidity cascades (chronic kidney disease, heart failure, COPD exacerbation), drug-disease interactions (hyperkalemia, hypoglycemia), and disease-progression pathways (ischemic stroke, sepsis, deep vein thrombosis). Prediction horizons are $\mathcal{T} = \{30, 90, 365\}$ days, which correspond to transitional care, follow-up planning for chronic disease, and longer-term risk stratification. We use a 70/10/20 split based on recorded discharge dates, with all admissions from a patient assigned to one partition. For an external site, we use the six one-year new-diagnosis tasks of EHRSHOT [37]. Appendix A details the cohorts.

Knowledge Graph. We use PrimeKG [5], which integrates 20 biomedical resources into approximately 129K nodes and 4.0M edges across 30 relation types. After disease-specific refinement (Section 2.2), the average evidence graph G_d contains approximately 5.2K nodes and 14.8K edges, with 45% of candidate edges removed by the support threshold and about 12% grouped as redundant (Appendix A).

Baselines. We compare against three categories of methods. EHR-only models use no external knowledge and include RETAIN [9], StageNet [13], and Med-BERT [32]. Static KG methods incorporate external knowledge as fixed features and include GRAM [8] and GraphCare [18]. Graph4LLM methods reason over the same evidence graph G_d without budget constraints, source-quality scoring, or citation-integrity training, and include ToG 2.0 [30], VoG [42], and KARE [19]. ToG-adapt does not verify retrieved edges, VoG-adapt verifies each step, and KARE-adapt retrieves community summaries. All LLM-based methods use GPT-4o-mini [16], and we also report BAR with Qwen3-8B [38]. Appendix B describes the adaptations.

Metrics. We report AUROC and AUPRC for each disease-horizon pair and take the macro-average across all 24 tasks, with AUPRC as the primary metric because positive outcomes are uncommon. Citation precision estimates the fraction of displayed citations that a GPT-4o judge labels relevant to the patient's risk pathway using a uniform sample of returned citations. On pairs where two annotators agree, the judge matches their label with a Cohen's κ of 0.83 (Appendix H). Evidence quality is the mean $\phi^\star$ of displayed citations, and both citation metrics exclude fallback episodes. Budget utilization is the share of the assigned budget that is spent, a reasoning step is one acquisition attempt including failed ones, and efficiency is measured in LLM tokens per episode. For clinical utility, we report the Brier score [4], the integrated calibration index (ICI) [2], and decision-curve net benefit [35] after recalibration on the validation split. Table 1 reports means and sample standard deviations over three independent training seeds on the same data split. Appendix B gives the remaining settings.

3.2 Main Results

BAR achieves the highest AUROC and AUPRC on both datasets (Table 1), with an AUPRC advantage of about three percentage points over the strongest baseline, which is ToG-adapt on MIMIC-III and

Table 1: Main results across eight diseases and three horizons over three independent training seeds. All metrics except steps are percentages. Bold marks BAR with GPT-4o-mini, and underlining marks the strongest non-BAR baseline for each prediction metric.

		MIMIC-III		MIMIC-IV	
Category	Method	AUROC	AUPRC	AUROC	AUPRC
EHR-only	RETAIN	$71.31_{\pm 0.38}$	$9.53_{\pm 0.21}$	$70.46_{\pm 0.48}$	$9.13_{\pm 0.49}$
	StageNet	$72.89_{\pm 0.36}$	$10.63_{\pm 0.36}$	$72.39_{\pm 0.26}$	$9.66_{\pm 0.24}$
	Med-BERT	$74.46_{\pm 0.43}$	$11.44_{\pm 0.25}$	$73.97_{\pm 0.94}$	$11.04_{\pm 0.44}$
Static KG	GRAM	$75.10_{\pm 0.59}$	$12.45_{\pm 0.22}$	$75.38_{\pm 0.61}$	$12.46_{\pm 0.30}$
	GraphCare	$77.34_{\pm 0.42}$	$14.29_{\pm 0.37}$	$76.88_{\pm 0.42}$	$13.69_{\pm 0.29}$
Graph4LLM	KARE-adapt	$78.49_{\pm 0.30}$	$16.06_{\pm 0.28}$	$77.77_{\pm 0.48}$	$\underline{15.88}_{\pm 0.17}$
	ToG-adapt	$78.61_{\pm 0.53}$	$\underline{16.20}_{\pm 0.51}$	$77.78_{\pm 0.32}$	$14.56_{\pm 0.23}$
	VoG-adapt	$\underline{79.21}_{\pm 0.70}$	$15.94_{\pm 0.29}$	$\underline{78.60}_{\pm 0.38}$	$15.41_{\pm 0.45}$
BAR (Ours)	Qwen3	$80.43_{\pm 0.27}$	$17.08_{\pm 0.87}$	$79.23_{\pm 0.54}$	$16.85_{\pm 0.62}$
	GPT-4o-mini	$\mathbf{82.41}_{\pm 0.65}$	$\mathbf{19.50}_{\pm 0.24}$	$\mathbf{81.90}_{\pm 0.19}$	$\mathbf{18.70}_{\pm 0.48}$

Reasoning Quality & Efficiency (MIMIC-III / MIMIC-IV)

Method	Citation Precision	Evidence Quality	Budget Utilization	Reasoning Steps	Fallback Rate
ToG-adapt	$53.21_{\pm 0.72}$ / $50.21_{\pm 1.00}$	$47.98_{\pm 0.73}$ / $46.22_{\pm 1.11}$	—	$6.65_{\pm 0.11}$ / $7.07_{\pm 0.12}$	—
VoG-adapt	$59.00_{\pm 1.45}$ / $56.63_{\pm 1.81}$	$48.20_{\pm 1.01}$ / $45.92_{\pm 1.74}$	—	$5.57_{\pm 0.12}$ / $6.01_{\pm 0.15}$	—
BAR (Qwen3)	$72.04_{\pm 0.86}$ / $64.55_{\pm 0.99}$	$56.78_{\pm 0.75}$ / $52.83_{\pm 1.22}$	$69.72_{\pm 1.96}$ / $70.54_{\pm 1.32}$	$4.38_{\pm 0.14}$ / $4.44_{\pm 0.09}$	$22.26_{\pm 0.92}$ / $27.04_{\pm 2.34}$
BAR (GPT-4o-mini)	$\mathbf{77.06}_{\pm 0.81}$ / $\mathbf{73.64}_{\pm 1.19}$	$\mathbf{64.84}_{\pm 1.74}$ / $\mathbf{60.86}_{\pm 0.86}$	$\mathbf{61.86}_{\pm 1.27}$ / $\mathbf{65.06}_{\pm 1.35}$	$\mathbf{3.83}_{\pm 0.09}$ / $\mathbf{4.05}_{\pm 0.09}$	$\mathbf{15.54}_{\pm 0.34}$ / $\mathbf{19.75}_{\pm 0.82}$

KARE-adapt on MIMIC-IV. BAR remains ahead with either LLM. Qwen3-8B gives lower prediction and citation quality than GPT-4o-mini but retains an advantage over the baselines. The relative AUPRC gain over the best EHR-only baseline grows at longer horizons (Appendix C). External evidence may help capture relationships that are less apparent in the discharge record, although outcome prevalence also varies with the horizon.

BAR combines higher citation precision with fewer acquisition attempts and fewer tokens. It uses about two thirds of the tokens of ToG-adapt and half those of VoG-adapt, mainly because each step shows the LLM only K_{pre} filtered candidates. Most episodes use several attempts, allowing the system to verify intermediate evidence and revise unsuccessful paths. ToG-adapt still makes fewer LLM calls because it omits verification (Table 7).

3.3 Ablation Study

Figure 2 compares each component removal with the full model. Removing the evidence graph causes the largest prediction loss, while several other components have a stronger effect on citation precision. One-shot retrieval reduces both metrics, and removing verification lowers AUPRC more than removing plan revision. The w/o paired gain variant continues stage 3 policy optimization with cost and citation-integrity rewards, while stage 2 uses uniform routing. Removing paired gain substantially lowers AUPRC, whereas removing citation integrity leaves prediction nearly unchanged but substantially reduces citation precision. At matched budgets, support-greedy navigation loses more AUPRC than removing verification, and random navigation falls below GraphCare (Appendix D.1).

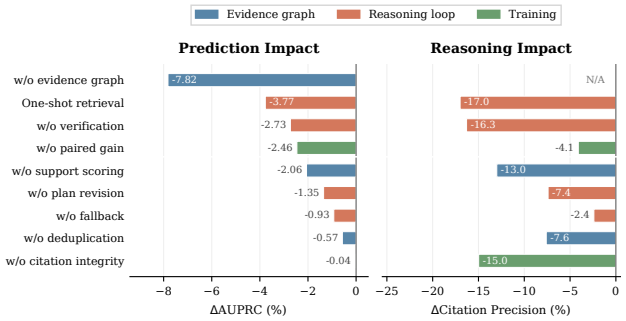

Figure 2: Ablation results from the full model, macro-averaged over tasks and datasets. Colors indicate framework stage.

3.4 Analysis

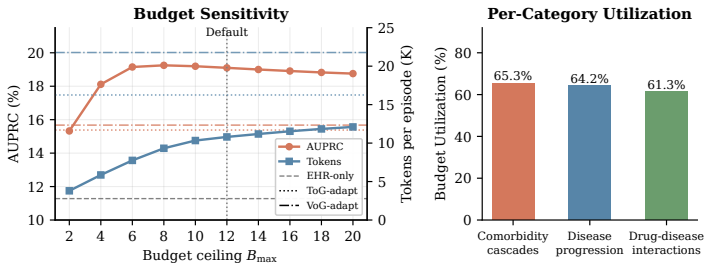

Figure 3: AUPRC and mean tokens per episode across budget ceilings, with ToG-adapt and VoG-adapt as reference lines (left), and per-category budget utilization (right), averaged over both datasets.

Budget Sensitivity. AUPRC rises quickly at low budgets, peaks at $B_{\max} = 8$ just below the validation-selected default, and declines slightly while token use keeps rising (Figure 3, left). From $B_{\max} = 4$ upward, BAR exceeds ToG-adapt and VoG-adapt in AUPRC, and it uses fewer tokens than both at every ceiling. Comorbidity cascades use a larger share of their assigned budgets than drug-disease interactions (Figure 3, right), and since every category follows the same allocation rule, this difference reflects how far the loop explores. A plausible clinical reading is that a drug-disease interaction is often supported by a direct drug-effect relation, whereas a comorbidity cascade passes through intermediate conditions and organ systems in larger evidence graphs (Table 5). Drug-disease interactions also fall back more often (Table 11), which suggests the loop stops early when a short pathway is missing or weakly supported.

Robustness. BAR's relative advantage over VoG-adapt grows as fewer ICD codes are retained. On held-out diseases, its relative decline is larger than GraphCare's but smaller than VoG-adapt's, while its absolute AUPRC remains higher than both (Appendix D). The advantage also holds across graph-construction settings, with UMLS in place of PrimeKG, under reduced linker coverage, and on EHRSHOT, and it narrows but persists when laboratory values and medications are added. Corrupting navigation candidates lowers AUPRC linearly while raising fallback (Appendices D and F).

Fallback Behavior. Fallback is more frequent on MIMIC-IV than on MIMIC-III (Table 1). Removing it lowers overall AUPRC, which supports an evidence threshold before graph information enters the prediction, at the cost of no citations for the episodes that fall back (Appendix C).

Lead Time. Because a diagnosis recorded within days of discharge may reflect a condition already present but not yet coded, we remove index admissions whose target disease is first recorded within seven days of discharge and retrain every method. AUPRC drops for every method by a similar fraction, and BAR remains strongest on both datasets. Its margin disappears at 30 days, where early events make up about a quarter of positives, and grows with the horizon, so external pathways add most for later outcomes (Appendix C.2).

Clinical Utility. Raw scores of BAR, KARE-adapt, and the stage-1 EHR-only model are similarly miscalibrated, and recalibration on the validation split lowers ICI about fourfold. After recalibration, BAR has the highest decision-curve net benefit at every threshold and horizon, and its relative lead grows as the threshold rises past the outcome prevalence (Appendix E).

3.5 Case Study

The patient in Figure 4 has atrial fibrillation. The initial query retrieves evidence for left atrial stasis and a hypercoagulable state. A proposed shortcut from stasis to cerebral embolism fails verification because the edge has weak support. The revised plan follows a pathway through left atrial thrombus and cardioembolism to ischemic stroke, retaining the previously accepted evidence. This expansion requires successive adjacency queries as new frontier nodes become available. Six edges are accepted, and the five highest-scoring edges form the final citation list within $K_C = 5$.

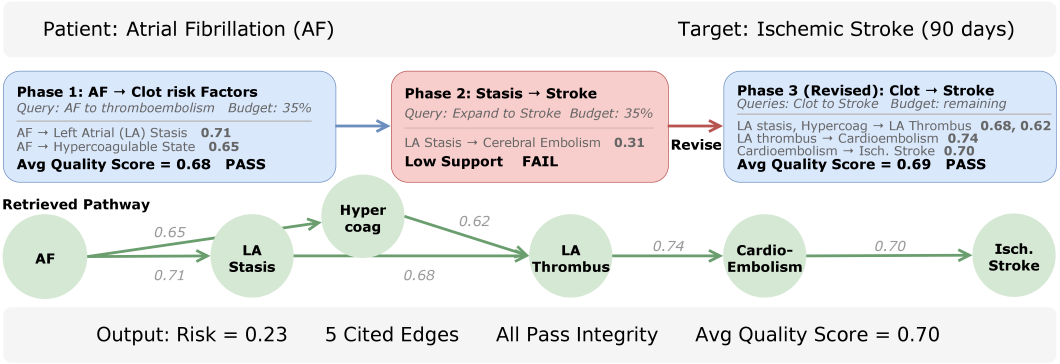

Figure 4: Reasoning trace for a patient with atrial fibrillation, from the initial plan through a failed step and its revision. Phase 3 groups at least three adjacency queries.

4 Related Work

Clinical Prediction from EHRs. Visit-sequence models such as RETAIN [9], Dipole [28], StageNet [13], GCT [10], AdaCare [29], and pretrained transformers [32, 25] learn temporal patterns from coded diagnoses and procedures but have no mechanism to recover risk pathways that are absent from the record. KG-augmented methods such as GRAM [8], CGL [26], GraphCare [18], and BoxLM [34] address this gap by incorporating relational knowledge from external graphs, yet they treat the graph as a source of embeddings extracted before prediction rather than as a reasoning space explored at inference time. KARE [19] moves closer by organizing a multi-source graph into summarized communities and retrieving community summaries for each patient before an LLM reasons toward a prediction. Because each summary condenses many relations, its predictions do not cite individual sourced edges, and its retrieval has no patient-specific acquisition budget.

LLM Reasoning over KGs. Structured multi-step reasoning [36, 39] and retrieval-augmented generation [24] have enabled LLMs to iteratively plan, retrieve, and verify over knowledge graphs [30, 27, 6, 12, 42]. These Graph4LLM methods assume an open-domain setting where exploration can run until the model chooses to stop, and all retrieved edges are treated uniformly. Cost-sensitive feature acquisition [17] introduces budget constraints but operates on flat feature vectors without graph structure. Adaptive retrieval [1, 20] decides when to retrieve based on expected utility but does not maintain source provenance across retrieval steps. RL-based graph traversal [11] learns exploration policies over graphs but does not differentiate edges by evidential backing. BAR addresses all three requirements jointly in a clinical prediction setting.

5 Conclusion and Limitations

We presented BAR, a framework for budget-aware LLM reasoning over medical knowledge graphs for post-discharge risk prediction. Across eight diseases and three prediction horizons on MIMIC-III and MIMIC-IV, BAR achieves higher prediction and citation quality than the compared baselines with fewer LLM tokens, and its gain over EHR-only prediction grows at longer horizons and carries over to an external benchmark. Its labels cover only diagnoses recorded within one health system and do not model censoring or competing risks, its raw scores need local recalibration, and the medical correctness of its full reasoning traces remains untested. Prospective clinical validation is needed to assess whether source-backed rationales improve clinician trust in practice.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Foundation under Grant No. OIA-2531881. We also thank the anonymous reviewers for their valuable suggestions and feedback. The authors declare no competing interests.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

[2] Peter C Austin and Ewout W Steyerberg. The Integrated Calibration Index (ICI) and related metrics for quantifying the calibration of logistic regression models. *Statistics in Medicine*, 38(21):4051–4065, 2019.

[3] Olivier Bodenreider. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270, 2004.

[4] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[5] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023.

[6] Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. Plan-on-Graph: Self-correcting adaptive planning of large language model on knowledge graphs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[7] Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[8] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. GRAM: Graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 787–795, 2017.

[9] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in Neural Information Processing Systems*, 29, 2016.

[10] Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 606–613, 2020.

[11] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*, 2018.

[12] Junqi Gao, Xiang Zou, Ying Ai, Dong Li, Yichen Niu, Biqing Qi, and Jianxing Liu. Graph Counselor: Adaptive graph exploration via multi-agent synergy to enhance LLM reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24650–24668, 2025.

[13] Junyi Gao, Cao Xiao, Yasha Wang, Wen Tang, Lucas M Glass, and Jimeng Sun. StageNet: Stage-aware neural networks for health risk prediction. In *Proceedings of The Web Conference 2020*, pages 530–540, 2020.

[14] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

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

[16] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[17] Jaromír Janisch, Tomáš Pevnỳ, and Viliam Lisỳ. Classification with costly features using deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3959–3966, 2019.

[18] Pengcheng Jiang, Cao Xiao, Adam Richard Cross, and Jimeng Sun. GraphCare: Enhancing healthcare predictions with personalized knowledge graphs. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Pengcheng Jiang, Cao Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025.

[20] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, 2023.

[21] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1, 2023.

[22] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

[23] Sunjun Kweon, Jiyoun Kim, Heeyoung Kwak, Dongchul Cha, Hangyul Yoon, Kwanghyun Kim, Jeewon Yang, Seunghyun Won, and Edward Choi. EHRNoteQA: An LLM benchmark for real-world clinical practice using discharge summaries. *Advances in Neural Information Processing Systems*, 37:124575–124611, 2024.

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[25] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. BEHRT: Transformer for electronic health records. *Scientific Reports*, 10(1):7155, 2020.

[26] Chang Lu, Chandan K Reddy, Prithwish Chakraborty, Samantha Kleinberg, and Yue Ning. Collaborative graph learning with auxiliary text for temporal event prediction in healthcare. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[27] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1903–1911, 2017.

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

[30] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-Graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1):18, 2018.

[32] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine*, 4(1):86, 2021.

[33] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

[34] Yanchao Tan, Hang Lv, Yunfei Zhan, Guofang Ma, Bo Xiong, and Carl Yang. BoxLM: Unifying structures and semantics of medical concepts for diagnosis prediction in healthcare. In *Forty-second International Conference on Machine Learning*, 2025.

[35] Andrew J Vickers and Elena B Elkin. Decision curve analysis: a novel method for evaluating prediction models. *Medical Decision Making*, 26(6):565–574, 2006.

[36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[37] Michael Wornow, Rahul Thapa, Ethan Steinberg, Jason Fries, and Nigam Shah. EHRSHOT: An EHR benchmark for few-shot evaluation of foundation models. *Advances in Neural Information Processing Systems*, 36:67125–67137, 2023.

[38] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[39] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36:11809–11822, 2023.

[40] Junzi Zhang, Jongho Kim, Brendan O'Donoghue, and Stephen Boyd. Sample efficient reinforcement learning with REINFORCE. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10887–10895, 2021.

[41] Yao Zhang, Hongru Liang, Adam Jatowt, Wenqiang Lei, Xin Wei, Ning Jiang, and Zhenglu Yang. GMH: A general multi-hop reasoning model for KG completion. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3437–3446, 2021.

[42] Wenxin Zhao, Jiachuan Wang, Yongqi Zhang, Shuangyin Li, Cheng Deng, Jun Wang, and Lei Chen. VoG: Enhancing LLM reasoning through stepwise verification on knowledge graphs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[43] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Appendix Contents

A Dataset and Knowledge Graph Statistics **15**

A.1 Prediction Cohorts 15

A.2 Lead-Time Cohorts 15

A.3 External Site 16

A.4 Evidence Graphs 17

B Implementation Details and Pseudocode **18**

B.1 Patient Representations and Knowledge Preparation 18

B.2 Inference and Budget Accounting 19

B.3 Training 19

B.4 Evaluation Protocol 21

C Detailed Results **22**

C.1 Per-Disease and Per-Horizon Results 22

C.2 Results under the Seven-Day Lead Time 22

D Robustness and Generalization **25**

D.1 Navigation Policy and Candidate Corruption 25

D.2 Input Sparsity and Disease Transfer 25

D.3 Knowledge Source and Linker Coverage 26

D.4 External Site 26

D.5 Laboratory and Medication Inputs 26

E Clinical Utility **27**

F Hyperparameter Sensitivity **27**

F.1 Reasoning Hyperparameters 27

F.2 Budget Ceiling 29

F.3 Graph Construction 29

G Evidence-Quality Score Analysis **29**

H Citation Precision Evaluation **30**

I LLM Prompt Templates **30**

J Broader Impact and Ethics **31**

A Dataset and Knowledge Graph Statistics

A.1 Prediction Cohorts

Index admissions. Candidate indices are adult hospital admissions with at least one linked ICU stay, valid admission and discharge times, and discharge alive. Hospital admissions are counted once, regardless of the number of linked ICU stays or diagnosis records. Prior history and subsequent outcomes include all hospital admissions in each patient's history. Hospital episodes do not overlap. For each disease, we exclude indices with the target diagnosis in the index admission or any earlier admission. All remaining indices are retained, and the same set is used at every prediction horizon. A patient may contribute more than one index admission for a disease.

Outcome timing. Hospital diagnosis codes describe a stay and are assigned for billing at discharge [22, 21]. We date an outcome by the discharge of the subsequent admission carrying the diagnosis. Let a_j and τ_j denote admission j's admission and discharge times, and let p_j identify its patient. For an eligible index i and disease d, the first recorded event time is

$$t^{\star}_{i,d} = \min\{\tau_j : p_j = p_i,\ a_j > \tau_i,\ j \text{ carries a diagnosis of } d\}, \tag{11}$$

with $t^{\star}_{i,d} = +\infty$ when no qualifying admission exists. Ties are resolved by admission identifier. The label is $y^{(h)}_{i,d} = \mathbb{1}[t^{\star}_{i,d} \le \tau_i + h]$. This timestamp identifies the recorded diagnosis rather than biological onset. A zero label denotes no qualifying recorded diagnosis within the window, including cases without a recorded readmission.

Counting units. For the eligible index set $\mathcal{I}_d$, the cohort size, positive count, and positive rate are

$$N_d = |\mathcal{I}_d|, \qquad P_d(h) = \sum_{i \in \mathcal{I}_d} y^{(h)}_{i,d}, \qquad \mathrm{Rate}_d(h) = 100\, P_d(h)/N_d. \tag{12}$$

Let $\mathcal{I}_0$ be the candidate index set before disease-specific exclusions, with $N_0 = |\mathcal{I}_0|$. The count $F_0(h)$ includes candidate indices with at least one hospital admission that starts after the index discharge and ends within h days. Applying the same rule to $\mathcal{I}_d$ gives $F_d(h)$. Each index is counted once, even with several returns. These counts satisfy $P_d(h) \le F_d(h) \le N_d$ and $F_d(h) \le F_0(h)$. For $F_d(h) > 0$, $q_d(h) = P_d(h)/F_d(h)$ is the proportion of returning indices with a positive target-disease label.

One later admission can label several earlier indices and more than one disease. Positive index–disease pairs, distinct positive indices, distinct positive patients, and distinct outcome admissions are therefore reported separately. A patient-disease event is counted once even when it labels several indices. An outcome admission is counted once across all target diseases. Patient-level counts take the union of positive labels over all eligible index-specific windows. These event counts include first recorded disease admissions that produce at least one positive label within the horizon.

Table 2 reports full-cohort counts before data partitioning. The denominator N_d includes indices without a recorded subsequent admission and is identical at 30, 90, and 365 days.

Cohort summary. The candidate cohorts include 33,997 patients in MIMIC-III and 45,195 in MIMIC-IV. Returning candidate indices number 4,418, 7,142, and 9,473 at 30, 90, and 365 days in MIMIC-III, and 9,343, 15,856, and 23,718 in MIMIC-IV. Table 2 reports the admission totals and deduplicated outcomes. Across diseases and horizons, $q_d(h)$ ranges from 6.13% to 33.16% in MIMIC-III and 5.34% to 42.25% in MIMIC-IV. At 365 days, each returning candidate index has a mean of 1.12 and 0.95 positive disease labels, respectively. At 30 days, $100F_d(30)/N_d$ ranges from 9.15% to 9.87% and from 10.79% to 14.05%.

Among positive patients, the counts with one, two, three, and at least four target diseases are 2,115, 1,259, 589, and 355 in MIMIC-III and 4,869, 2,924, 1,378, and 870 in MIMIC-IV. The corresponding means are 1.84 and 1.86 diseases per positive patient, and 1.34 and 1.21 positive indices per patient-disease event.

A.2 Lead-Time Cohorts

The seven-day definition removes every index whose first recorded event $t^{\star}_{i,d}$ falls in $(\tau_i, \tau_i + 7]$. These indices are removed instead of relabeled, because a patient diagnosed two days after discharge

Table 2: Cohort accounting. N_d counts eligible index admissions. Each horizon reports the positive count $P_d(h)$ with its percentage of N_d in parentheses, followed by the count of indices with a completed hospital return $F_d(h)$. The bottom panel reports the deduplicated 365-day totals.

MIMIC-III

		30 days		90 days		365 days	
Disease	N_d	P_d (%)	F_d	P_d (%)	F_d	P_d (%)	F_d
CKD	35,147	524 (1.49)	3,309	962 (2.74)	5,362	1,593 (4.53)	7,123
HF	31,173	595 (1.91)	2,853	1,129 (3.62)	4,607	1,690 (5.42)	6,106
COPD	41,114	294 (0.72)	4,038	555 (1.35)	6,529	850 (2.07)	8,654
HyperK	38,165	585 (1.53)	3,623	1,116 (2.92)	5,892	1,624 (4.26)	7,796
HypoG	42,014	254 (0.60)	4,146	473 (1.13)	6,696	689 (1.64)	8,886
Stroke	40,325	266 (0.66)	3,921	518 (1.28)	6,357	820 (2.03)	8,437
Sepsis	35,423	960 (2.71)	3,310	1,670 (4.71)	5,286	2,316 (6.54)	6,985
DVT	41,391	449 (1.08)	4,065	763 (1.84)	6,538	1,059 (2.56)	8,683

MIMIC-IV

		30 days		90 days		365 days	
Disease	N_d	P_d (%)	F_d	P_d (%)	F_d	P_d (%)	F_d
CKD	39,584	726 (1.83)	4,760	1,516 (3.83)	8,084	3,221 (8.14)	12,161
HF	36,050	766 (2.12)	4,236	1,563 (4.34)	7,120	3,183 (8.83)	10,722
COPD	54,504	463 (0.85)	7,517	977 (1.79)	12,797	1,927 (3.54)	19,215
HyperK	44,466	926 (2.08)	5,492	1,842 (4.14)	9,328	3,447 (7.75)	14,091
HypoG	56,089	421 (0.75)	7,882	876 (1.56)	13,381	1,707 (3.04)	19,996
Stroke	51,917	501 (0.97)	7,035	1,021 (1.97)	11,994	2,095 (4.04)	18,021
Sepsis	36,184	1,215 (3.36)	3,906	2,373 (6.56)	6,572	4,232 (11.70)	10,016
DVT	53,096	829 (1.56)	7,222	1,621 (3.05)	12,281	2,806 (5.28)	18,394

Dataset-level counts at 365 days

Count	MIMIC-III	MIMIC-IV
Candidate index admissions N_0	43,966	62,815
Indices with a completed return $F_0(365)$	9,473	23,718
Positive index–disease pairs $\sum_d P_d(365)$	10,641	22,618
Distinct positive indices	6,116	13,486
Distinct positive patients	4,318	10,041
Distinct patient–disease events	7,955	18,711
Distinct outcome admissions	5,441	14,256

is not a clean negative at any horizon. The remaining positives at horizon h are the indices with $t^{\star}_{i,d} \in (\tau_i + 7, \tau_i + h]$. Every method is retrained on the reduced cohorts with the same splits and settings. Table 3 reports the counts. The removed indices are under 1% of each cohort, but they make up 28% and 25% of 30-day positives in MIMIC-III and MIMIC-IV, against 23% if events were spread evenly over 30 days. Their share falls to 15% and 13% at 90 days and to 10% and 7% at 365 days, so the definition mainly changes the 30-day task.

A.3 External Site

EHRSHOT [37] contains longitudinal records of 6,739 Stanford Medicine patients in the OMOP common data model, covering care beyond the ICU. Its six new-diagnosis tasks ask, at discharge from an inpatient visit, whether a first diagnosis of hypertension, hyperlipidemia, acute myocardial infarction, pancreatic cancer, celiac disease, or lupus will be recorded within one year. We use the patient-level splits and task labels distributed with the public EHRSHOT benchmark release. All methods use diagnosis concepts recorded up to the prediction time. Patient inputs exclude laboratory values, medications, procedures, and clinical notes. Diagnosis concepts are mapped to PrimeKG through UMLS, and an evidence graph is built for each target with the same PrimeKG snapshot and construction settings. The reasoning policy and prompts are reused, and only the EHR encoder and prediction heads are trained on the EHRSHOT training split. Med-BERT, VoG-adapt, and

Table 3: Cohort counts under the seven-day lead time. $P_d(7)$ counts indices whose first recorded event falls within seven days of discharge and are removed. $N_d^{(7)}$ is the remaining cohort and $P_h^{(7)}$ its positive count at h days.

Data	Disease	$P_d(7)$	$N_d^{(7)}$	$P_{30}^{(7)}$	$P_{90}^{(7)}$	$P_{365}^{(7)}$
MIMIC-III	CKD	93	35,054	431	869	1,500
	HF	148	31,025	447	981	1,542
	COPD	64	41,050	230	491	786
	HyperK	212	37,953	373	904	1,412
	HypoG	76	41,938	178	397	613
	Stroke	51	40,274	215	467	769
	Sepsis	293	35,130	667	1,377	2,023
	DVT	152	41,239	297	611	907
MIMIC-IV	CKD	118	39,466	608	1,398	3,103
	HF	169	35,881	597	1,394	3,014
	COPD	91	54,413	372	886	1,836
	HyperK	313	44,153	613	1,529	3,134
	HypoG	119	55,970	302	757	1,588
	Stroke	103	51,814	398	918	1,992
	Sepsis	346	35,838	869	2,027	3,886
	DVT	220	52,876	609	1,401	2,586

KARE-adapt follow the same protocol. Positive rates in the test split range from about 13% for hypertension and hyperlipidemia to under 1% for celiac disease and lupus (Table 4). Appendix D.4 reports the results.

Table 4: Test split of the public EHRSHOT benchmark release. A patient can contribute more than one label, and the rate is the share of positive labels.

Task	Patients	Labels	Positive	Rate (%)
Hypertension	755	1,258	159	12.64
Hyperlipidemia	864	1,317	172	13.06
Acute myocardial infarction	1,235	2,127	144	6.77
Pancreatic cancer	1,246	2,220	56	2.52
Celiac disease	1,255	2,222	21	0.95
Lupus	1,249	2,243	20	0.89

A.4 Evidence Graphs

Table 5 summarizes the refined evidence graphs. Appendix B.1 details support scoring and expansion-guide construction.

Table 5: Evidence graphs refined from PrimeKG [5] ($\sim$129K nodes and 4.0M edges).

Target disease	Nodes	Edges	Support removed (%)	Equivalence reduction (%)	Guide coverage (%)
Chronic kidney disease	6,841	18,273	42.3	13.7	87.4
Heart failure	7,214	19,847	41.8	14.2	89.1
COPD exacerbation	4,923	13,516	46.1	11.3	84.6
Hyperkalemia	3,847	10,284	48.7	10.8	82.3
Hypoglycemia	3,612	9,738	49.2	9.6	80.7
Ischemic stroke	5,731	16,142	43.6	12.9	86.8
Sepsis	6,183	17,421	44.1	13.1	88.2
Deep vein thrombosis	3,294	12,847	45.8	11.4	85.3
Average	*5,206*	*14,759*	*45.2*	*12.1*	*85.6*

B Implementation Details and Pseudocode

Table 6 summarizes the configuration. Algorithm 1 gives the inference procedure.

Table 6: Hyperparameters and configuration.

Evidence Graph		*Reasoning Loop*	
Hop limit K	3	Budget $B_{\min}$ / $B_{\max}$	2 / 12
Degree cap $d_{\max}$	50	Query / expand / select cost	1.0 / 0.5 / 0.5
Hub edge cap M	10	Pre-filter cap K_{pre}	5
Support threshold τ_s	0.3	Citation cap K_C	5
Min edges $K_{\min}$	2	Stop ϵ_{stop} / patience P_{stop}	0.01 / 2
Quality threshold τ_{fb}	0.4	Evidence cap K_E	8
		Temperature β	2.0
Model		*Training*	
Default LLM	GPT-4o-mini	Stage-1 epochs	30
Second LLM	Qwen3-8B	Stage-2 / stage-3 rounds	10 / 20
EHR encoder	2-layer GRU	Episodes per round	$\leq 4{,}096$
Dims (m, hidden)	256, 256	Predictor / policy LR	10^{-4} / 10^{-5}
Fusion $f(\cdot)$	2-layer MLP	Batch size	32
		$(\lambda_{\text{gain}}, \lambda_{\text{cost}}, \lambda_{\text{qual}})$	(1.0, 0.3, 0.5)
		Hardware	3×L40S (48GB each)

B.1 Patient Representations and Knowledge Preparation

Patient encoding. We represent each admission by the mean of its ICD code embeddings, which have dimension 256 and use Xavier uniform initialization. The visits in H_i are ordered chronologically, with the index admission X_i appended as the final visit. A two-layer GRU encodes this sequence, and its final hidden state gives $\mathbf{z}_i^{\text{ehr}} \in \mathbb{R}^{256}$. For patients without prior admissions, the sequence contains only the index admission.

ICD-to-KG linking. We normalize code punctuation while preserving leading zeros and the ICD version. Each code is joined to UMLS `MRCONSO.RRF` through the corresponding `CODE` and `SAB` fields, using `ICD9CM` or `ICD10CM`, to obtain distinct CUIs [3]. Both join sides share this normalization.

PrimeKG's `umls.py` and `map_umls_mondo.py` workflow produces the CUI-to-MONDO table `umls_mondo.csv` [5]. We join its MONDO identifiers to disease nodes in the selected PrimeKG export, respecting that export's grouping of MONDO concepts. Duplicate paths to the same node are collapsed. A code becomes an anchor only when it resolves to one distinct graph node. Unmapped and ambiguous codes are omitted from the anchor set. The resulting $\text{link}(c)$ is single-valued and fixed during training. Other node types remain available during graph traversal.

Edge representation. For each edge $e = (u, r, v) \in E_d$, the learned representation $\mathbf{v}(e) \in \mathbb{R}^{256}$ is computed as $\mathbf{v}(e) = \text{MLP}_{\text{edge}}([\mathbf{e}_u; \mathbf{e}_r; \mathbf{e}_v])$, where $\mathbf{e}_u, \mathbf{e}_v \in \mathbb{R}^{256}$ are the concept node embeddings from Section 2.3.1, $\mathbf{e}_r \in \mathbb{R}^{256}$ is a learnable relation-type embedding, and MLP_{edge} is a single-layer projection from $\mathbb{R}^{768}$ to $\mathbb{R}^{256}$ with ReLU activation.

Knowledge sources. We use fixed UMLS, MONDO, PrimeKG, and PubMed snapshots across all partitions. The resource manifest records versions, the PrimeKG checksum, mapping-script revision, and corpus date. Each edge stores its source identifier, version, and record key, plus PMIDs and dates for supporting abstracts. Anchor frequencies use only training records. This evaluation restricts EHR inputs to discharge but uses fixed external knowledge. Resource dates are not compared with patient-shifted admission dates.

Evidence support. The score $s(e) \in [0, 1]$ combines curated-source support with literature alignment. A source table distinguishes expert-curated records from computationally derived associations in PrimeKG's 20 constituent resources [5]. We count distinct curated resources linked to an edge,

excluding duplicate exports and reverse copies. Missing source attribution contributes zero. The curated score is

$$s_{\text{curated}}(e) = \min\left(1, \frac{n_{\text{curated}}(e)}{3}\right). \tag{13}$$

Across the evidence graphs, 91.4% of edges have $n_{\text{curated}} \leq 2$.

For literature support, we query the fixed PubMed abstract corpus with BM25 using the concatenated endpoint concept names. We retrieve the top three abstracts and compute PubMedBERT [14] cosine similarities between the concept-name pair and sentences in these abstracts. Similarities are clipped to $[0, 1]$ and averaged to obtain $s_{\text{text}}(e)$. The final score is

$$s(e) = 0.4\, s_{\text{curated}}(e) + 0.6\, s_{\text{text}}(e). \tag{14}$$

Both weights remain fixed. Appendix F reports sensitivity to the curated weight w_c and threshold τ_s.

Expansion guide. For disease d, an anchor is frequent when its ICD code appears in at least 1% of training admissions. The guide I_d stores its top 20 incident edge identifiers ranked by $s(e)$. At inference, QUERYGUIDE follows Section 2.3.4, with $K_{\text{pre}} = 5$ candidates passed to the LLM.

B.2 Inference and Budget Accounting

Inference accounting. Table 7 counts planning, selection, verification, revision, and retry calls per test episode. Every episode includes an initial plan, and ToG-adapt has no verification calls. A reasoning step is one executed acquisition attempt, including failures. The first attempt uses QUERYGUIDE plus selection and costs 1.5 units. Later attempts use QUERYEXPAND plus selection and cost 1.0 unit each. Empty selections and failed verification remain charged. Zero-attempt episodes incur no acquisition cost. Planning, verification, and revision count as LLM calls, while transport retries add calls without another acquisition charge. Affordability is checked before each attempt.

Budgets follow Eq. (4) using the task-specific training uncertainty CDF, held fixed for validation and testing. On the evaluated episodes, $\overline{B} = 2 + 10\overline{u}$. Within each run, Budget Utilization is $100 \sum_i C_i / \sum_i B_i$, including assigned budgets for zero-attempt episodes. Table 1 reports its mean and sample SD across runs. The metric is undefined for unbudgeted baselines.

For category g, let $B_g = \sum_{i \in g} B_i$ and $U_g = 100 \sum_{i \in g} C_i / B_g$. Overall utilization is $\sum_g B_g U_g / \sum_g B_g$. These weights reduce to episode-count weights when categories have equal mean assigned budgets.

Table 7: LLM calls and acquisition steps per test episode on MIMIC-III, averaged over three runs. Totals use unrounded counts.

Method	Plan	Select	Verify	Revise	Retry	Total	Steps
ToG-adapt	1.00	6.65	0.00	0.00	0.25	7.90	6.65
VoG-adapt	1.00	5.57	5.57	0.70	0.21	13.04	5.57
BAR	1.00	3.83	3.83	0.56	0.15	9.37	3.83
BAR (Qwen3)	1.00	4.38	4.38	0.79	0.17	10.73	4.38

Table 8 counts tokens over all LLM calls in an episode. BAR sends far fewer input tokens than ToG-adapt or VoG-adapt and more output tokens, because its plans and verifications are longer while each step shows only K_{pre} filtered candidates. The ratios are nearly the same on both datasets.

Most episodes include at least three acquisition attempts (Table 9).

B.3 Training

Initialization and sampling. Learnable embeddings use Xavier uniform initialization. The LLM uses temperature 0.0 for plan generation and verification, and 0.7 for exploration during training. Stage 1 uses all eligible training indices for 30 epochs. Stages 2 and 3 use 10 and 20 sampling rounds, respectively. Each round draws up to 4,096 eligible training indices per dataset uniformly without replacement, with a new draw in each round. Each selected index contributes one episode,

Algorithm 1 Budget-aware reasoning loop.

Require: Patient EHR (X_i, H_i), target (d, h), evidence graph G_d, EHR-only score $\hat{y}^{\mathrm{ehr}}_{i,d,h}$, max budget $B_{\max}$
Ensure: Risk score $\hat{y}^{(h)}_{i,d}$, citation list $\mathcal{C}$

$A_{i,d} \leftarrow \{\mathrm{link}(c) : c \in X_i,\ \mathrm{link}(c) \in V_d\}$ {anchor set}
$\mathbf{p}^{(0)} \leftarrow f_{\mathrm{state}}([\mathbf{z}^{\mathrm{ehr}}_i;\ \mathbf{u}(d,h)])$; $E \leftarrow \emptyset$; $C \leftarrow 0$
$u_i \leftarrow F^{\mathrm{train}}_{d,h}\big(1 - \big|2\hat{y}^{\mathrm{ehr}}_{i,d,h} - 1\big|\big)$; $B_i \leftarrow B_{\min} + (B_{\max} - B_{\min})u_i$
$P = [s_1, \ldots, s_n] \leftarrow$ LLM-PLAN$(X_i, H_i, d, h, A_{i,d}, B_i)$ {plan generation}
for each next step s_t in the current plan P **do**
  if the next acquisition and selection attempt is not affordable **then**
    break
  end if
  Resolve q_t to affordable query actions $\mathcal{Q}_t$ {query policy, Appendix B.3}
  if $\mathcal{Q}_t = \emptyset$ **then**
    break
  end if
  $a_t \leftarrow \arg\max_{a \in \mathcal{Q}_t} \pi_\theta(a \mid S_t)$; retrieve candidates using a_t
  Score: $\phi_t(e) \leftarrow s(e) \cdot \sigma(\beta \cdot \max\{\mathbf{p}^{(t-1)} \cdot \mathbf{e}_u,\ \mathbf{p}^{(t-1)} \cdot \mathbf{e}_v\})$; keep top-$K_{\mathrm{pre}}$, remove seen groups
  $E_t \leftarrow$ LLM-SELECT(candidates, $K_E - |E|$); $C \leftarrow C + c_t$
  result $\leftarrow$ LLM-VERIFY$(s_t, E_t, \hat{f}_t)$ {verification}
  if result $=$ `FAIL` **then**
    $\mathbf{p}^{(t)} \leftarrow \mathbf{p}^{(t-1)}$
    if a further acquisition and selection attempt is affordable **then**
      $P_{t+1:} \leftarrow$ LLM-REVISE$(s_t, E_t, E, B_i - C)$
    end if
  else
    $E \leftarrow E \cup E_t$; $\mathbf{p}^{(t)} \leftarrow \mathrm{GRU}(\mathbf{p}^{(t-1)}, \bar{\mathbf{v}}^{(t)})$
  end if
  if no further acquisition and selection attempt is affordable **or** $|E| \geq K_E$ **then**
    break {budget exhausted or evidence limit}
  else if $\Delta U_t < \epsilon_{\mathrm{stop}}$ for P_{stop} consecutive steps **then**
    break {quality stagnation}
  end if
end for
if $|E| < K_{\min}$ **or** $U_{\mathrm{final}} < \tau_{\mathrm{fb}}$ **then**
  return $\hat{y}^{\mathrm{ehr}}_{i,d,h}, \emptyset$ {frozen EHR prediction and no citations}
else
  $\mathbf{z}^{\mathrm{kg}} \leftarrow \sum_{e \in E} w(e)\, \mathbf{v}(e)$
end if
$\hat{y}^{(h)}_{i,d} \leftarrow \sigma\big(\mathbf{t}^\top_{d,h} f([\mathbf{z}^{\mathrm{ehr}}_i;\ \mathbf{z}^{\mathrm{kg}}]) + b_{d,h}\big)$
$\mathcal{C} \leftarrow$ top-K_C edges in E by $\phi^\star(e)$, dedup by $g(e)$
return $\hat{y}^{(h)}_{i,d},\ \mathcal{C}$

Table 8: Mean input, output, and total LLM tokens per test episode at the default setting.

	MIMIC-III			MIMIC-IV		
Method	Input	Output	Total	Input	Output	Total
ToG-adapt	13,841	1,517	15,358	15,541	1,607	17,148
VoG-adapt	18,420	1,954	20,374	21,076	2,101	23,177
BAR	7,935	2,341	10,276	8,873	2,452	11,325

Table 9: Reasoning-step distribution (%), pooling episodes across three runs.

Dataset	Method	1 step (%)	2 steps (%)	≥ 3 steps (%)
MIMIC-III	BAR	7.47	23.43	69.11
MIMIC-III	BAR (Qwen3)	7.45	26.59	65.96
MIMIC-IV	BAR	5.94	25.75	68.31
MIMIC-IV	BAR (Qwen3)	6.21	26.77	67.02

with a disease sampled uniformly from its eligible disease set and a horizon sampled uniformly from $\{30, 90, 365\}$ days. The stage 2 draws are shared across LLM-based methods within each seed. Stages 2 and 3 retain the final checkpoint of each stage. For each completed configuration and seed, full validation predictions are generated once and reused for hyperparameter selection and recalibration.

Evaluation calls. Offline API evaluation uses Batch requests, with dependent reasoning steps submitted after their preceding outputs are available. Responses to deterministic calls are reused only when the model version, prompt, complete inputs, and decoding settings match. Training exploration uses fresh calls, and stage 3 collects fresh on-policy trajectories.

Policy optimization details. The routing policy learns which concrete graph query to execute. The LLM generates and revises the plan, selects returned edges, and verifies evidence. Its weights stay frozen. At step t, the routing state S_t contains the current patient state $\mathbf{p}_{i,d,h}^{(t-1)}$, accepted evidence, current query intent q_t, target (d, h), and remaining budget $B_i - C_{t-1}$.

The query intent is resolved to allowed node and relation identifiers in G_d. The initial action set contains QUERYGUIDE calls on allowed anchors. Later sets contain QUERYEXPAND(x, r) calls on allowed frontier nodes and incident relation types. A frontier node is an anchor or an endpoint of accepted evidence. Invalid identifiers, queries with no unused admissible edges, and actions whose query-plus-selection cost exceeds the remaining budget are masked out. Let $\mathcal{Q}_t$ denote the remaining admissible action set. An empty action set ends the episode. Each selected action is one acquisition attempt.

A two-layer MLP with a 256-unit ReLU hidden layer assigns a scalar logit

$$\ell_\theta(S_t, a) = \mathrm{MLP}_\theta\Big([\mathbf{p}_{i,d,h}^{(t-1)}; \mathbf{e}_x; \mathbf{e}_r; \mathbf{u}(d, h); (B_i - C_{t-1})/B_i]\Big). \tag{15}$$

For QUERYGUIDE, x is the anchor and the relation feature is zero. For QUERYEXPAND, x and r identify the frontier node and relation. The action distribution is

$$\pi_\theta(a \mid S_t) = \frac{\exp \ell_\theta(S_t, a)}{\sum_{a' \in \mathcal{Q}_t} \exp \ell_\theta(S_t, a')}, \qquad a \in \mathcal{Q}_t. \tag{16}$$

Stage 2 uses a uniform distribution over the same admissible action sets. Stage 3 samples from π_θ, and evaluation selects the highest-probability action, with ties resolved by stable node and relation identifiers. After a query, candidate scoring, top-K_{pre} filtering, LLM edge selection, and verification follow Section 2.3.4.

Here θ contains the routing MLP parameters. The EHR encoder, graph embeddings, patient-state update, and predictor have parameters ψ and receive supervised updates. Within each policy batch, ψ and all LLM outputs are held fixed for the routing update. The rollout stores the log probability of every sampled action, including failed attempts. For a batch of n episodes, the policy loss is

$$\mathcal{L}_{\mathrm{policy}}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \mathrm{stopgrad}(r_{i,d_i,h_i} - b_k) \sum_{t=1}^{T_i} \log \pi_\theta(a_{i,t} \mid S_{i,t}). \tag{17}$$

The return is the undiscounted episode reward in Eq. (10). The baseline b_k is the moving average from earlier batches, initialized to zero and updated after the current policy step as $b_{k+1} = 0.99 b_k + 0.01 \bar{r}_k$, where $\bar{r}_k$ is the mean episode reward in batch k. The routing update uses the policy learning rate in Table 6. A separate binary cross-entropy update uses the predictor learning rate for ψ. Gradients from the policy loss do not pass through the reward, discrete graph lookups, or LLM calls.

B.4 Evaluation Protocol

Aggregation and uncertainty. Results are averaged over three independent training seeds, with the same seeds and data splits used across methods. Within each run, AUPRC is macro-averaged over diseases and horizons. Table 1 also reports sample SDs. Per-disease, per-horizon, efficiency, and clinical-utility summaries reuse the main runs. Auxiliary experiments use matched reference configurations, with gains and drops computed within each seed and dataset before averaging.

Identical configurations reuse stored predictions within each seed. EHRSHOT intervals use paired patient bootstrap with trained models held fixed. All aggregates use unrounded values.

Citation precision uses the sampled citations described in Appendix H. Evidence quality pools all returned citations within each run. Fallback episodes contribute to prediction, fallback, and coverage statistics. Coverage is the fraction of episodes with citations and equals one minus the fallback rate. Category and overall fallback rates pool episode counts within each run before averaging across seeds.

Baseline adaptation. EHR-only baselines (RETAIN, StageNet, Med-BERT) receive the same ICD codes and prediction targets. Static KG baselines (GRAM, GraphCare) use PrimeKG [5] as their knowledge source. GRAM learns attention-based code representations using the disease hierarchy extracted from PrimeKG. GraphCare constructs personalized patient-KG subgraphs from PrimeKG and applies graph neural networks to obtain patient representations. Both use the full PrimeKG rather than the refined evidence graph G_d, following their original designs.

ToG-adapt and VoG-adapt share the refined evidence graph G_d, GPT-4o-mini, and the ICD-to-KG linker with BAR. ToG-adapt follows the iterative beam search of Think-on-Graph 2.0 [30], where the LLM generates queries, retrieves candidate edges, and selects among them for multiple rounds. VoG-adapt follows the stepwise verification protocol of VoG [42], where the LLM verifies each retrieved edge against its hypothesis before proceeding. Neither method imposes a budget constraint, so both run until the LLM produces a stop signal or a maximum of 10 reasoning steps is reached. Table 7 reports their call and step counts. Neither method uses support-score differentiation during acquisition or citation-integrity training. Both use the same EHR representation and predictor as BAR (Section 2.4.1), with uniform edge pooling in place of quality-weighted pooling.

KARE-adapt follows KARE [19], which groups a medical KG into hierarchical communities, summarizes each community with an LLM, and retrieves the summaries most relevant to a patient. We run its community detection and summarization on the same evidence graph G_d with GPT-4o-mini and retrieve summaries for the concepts that the shared ICD-to-KG linker returns. The retrieved summaries feed the same EHR representation and predictor head as the other KG-augmented methods. This replaces KARE's original multi-source graph and fine-tuned LLM predictor, so the comparison isolates the retrieval strategy. KARE-adapt has no acquisition budget, and its citations are community summaries.

All baselines use the same binary cross-entropy loss and patient-disjoint splits. EHR-only and static KG baselines use the full training cohorts for 30 and 40 epochs, respectively. ToG-adapt, VoG-adapt, and KARE-adapt use 30 epochs of EHR-only warm-up followed by the same 10 sampled supervised rounds as stage 2 of BAR.

C Detailed Results

C.1 Per-Disease and Per-Horizon Results

Per-disease performance. BAR has the highest AUPRC on every disease except sepsis, where KARE-adapt is slightly ahead on both datasets (Table 10). The gains vary within each disease category, and no single baseline leads across tasks.

Prediction horizon. The relative gain over the strongest EHR-only baseline grows with the prediction horizon (Figure 5).

Fallback and coverage. Fallback is more frequent for drug-disease interactions than for comorbidity cascades (Table 11). Citation coverage is lower on MIMIC-IV, which falls back more often.

C.2 Results under the Seven-Day Lead Time

Every method keeps 76–85% of its primary macro AUPRC under the lead-time definition, and the EHR-only, static KG, Graph4LLM, and BAR groups keep their AUPRC order (Table 12). KARE-adapt has the highest baseline AUPRC on both datasets, narrowly ahead of ToG-adapt on MIMIC-III and VoG-adapt on MIMIC-IV, while VoG-adapt has the highest baseline AUROC. Against the

Table 10: AUPRC (%) per disease, averaged over three horizons within each run and then over three runs. Avg matches the mean reported in Table 1. Bold marks the best value in each column.

Method	CKD	HF	COPD	HyperK	HypoG	Stroke	Sepsis	DVT	Avg
MIMIC-III									
RETAIN	10.69	12.05	8.84	12.00	8.58	6.02	11.43	6.64	9.53
StageNet	12.08	12.79	9.90	13.71	8.99	6.62	12.98	8.00	10.63
Med-BERT	13.42	13.70	10.53	14.71	10.57	6.82	13.33	8.42	11.44
GRAM	14.52	14.89	11.71	15.41	10.83	8.08	14.39	9.77	12.45
GraphCare	16.44	17.14	13.08	17.41	12.41	9.26	16.15	12.44	14.29
ToG-adapt	18.06	19.71	14.30	19.83	14.07	10.94	18.46	14.21	16.20
VoG-adapt	17.97	18.70	14.82	18.23	13.89	10.72	18.71	14.51	15.94
KARE-adapt	16.22	14.80	16.43	19.81	14.59	12.20	**22.09**	12.31	16.06
BAR	**22.56**	**23.11**	**17.92**	**23.34**	**16.92**	**13.55**	22.07	**16.55**	**19.50**
MIMIC-IV									
RETAIN	10.20	11.52	7.90	12.09	8.28	5.50	11.20	6.31	9.13
StageNet	11.78	11.83	8.10	12.29	8.46	6.12	12.01	6.68	9.66
Med-BERT	12.99	13.43	10.63	14.22	9.23	6.23	12.80	8.83	11.04
GRAM	16.10	16.35	11.05	15.15	9.66	6.97	13.70	10.70	12.46
GraphCare	15.45	16.52	11.97	16.87	12.23	8.20	15.25	13.04	13.69
ToG-adapt	16.41	17.03	13.36	18.09	13.27	9.42	17.04	11.86	14.56
VoG-adapt	18.07	18.81	13.55	17.19	14.30	9.63	17.46	14.28	15.41
KARE-adapt	14.98	18.29	14.49	19.99	15.16	9.31	**22.40**	12.45	15.88
BAR	**21.20**	**22.41**	**16.33**	**22.05**	**16.32**	**12.62**	22.00	**16.64**	**18.70**

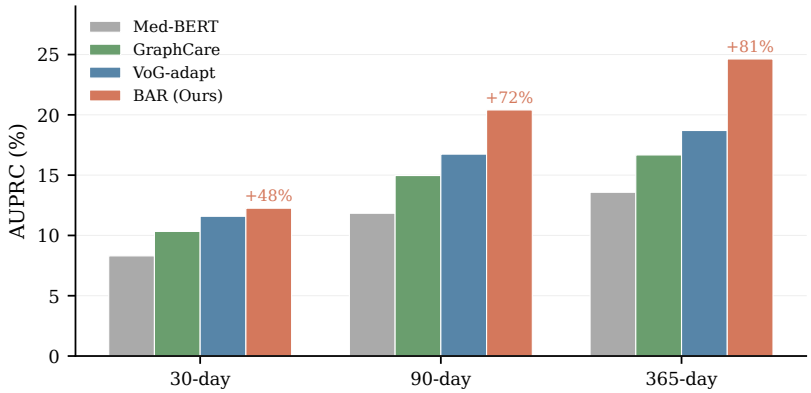

Figure 5: Per-horizon AUPRC averaged across both datasets. Percentages show the relative gain of BAR over the best EHR-only baseline.

Table 11: Fallback rates and citation coverage (%). Category rates pool episode counts within each run before averaging runs.

	MIMIC-III	MIMIC-IV
Fallback rate (%)		
Comorbidity cascades	11.42	15.46
Disease progression	14.11	18.40
Drug-disease interactions	22.58	26.84
Overall fallback (%)	15.54	19.75
Citation coverage (%)	84.46	80.25

baselines, BAR leads on seven of eight diseases on MIMIC-III and six on MIMIC-IV (Table 13). It ties KARE-adapt on COPD in MIMIC-IV and trails it on sepsis on both datasets, as under the primary definition. By horizon, the margin over the strongest comparator stays within 0.4 points at 30 days, slightly negative on MIMIC-IV, and grows at 90 and 365 days (Table 14). The 30-day task loses about a quarter of its positives to the lead time, and it is also the horizon where external evidence adds least under the primary definition.

Table 12: Macro AUROC and AUPRC (%) under the seven-day lead-time definition.

		MIMIC-III		MIMIC-IV	
Category	Method	AUROC	AUPRC	AUROC	AUPRC
EHR-only	RETAIN	69.84	7.60	68.90	7.49
	StageNet	71.79	8.77	70.78	7.90
	Med-BERT	73.03	9.09	72.51	9.03
Static KG	GRAM	72.71	9.42	73.45	9.53
	GraphCare	76.15	11.71	75.52	11.26
Graph4LLM	ToG-adapt	77.30	13.20	76.28	11.92
	VoG-adapt	77.76	12.65	77.56	12.95
	KARE-adapt	77.39	13.34	76.14	12.97
BAR	Qwen3	79.39	14.16	78.15	14.28
	GPT-4o-mini	**80.94**	**15.75**	**80.42**	**15.43**

Table 13: AUPRC (%) by disease under the seven-day lead time, averaged over the three horizons. Bold marks the best value in each column.

Method	CKD	HF	COPD	HyperK	HypoG	Stroke	Sepsis	DVT
MIMIC-III								
RETAIN	8.71	9.15	8.14	9.32	6.74	5.27	8.96	4.54
StageNet	10.40	10.67	8.40	11.88	7.53	5.74	10.12	5.39
Med-BERT	11.96	10.85	8.83	10.42	8.21	5.57	10.54	6.32
GRAM	12.74	10.92	9.70	10.69	7.51	6.33	10.36	7.11
GraphCare	14.51	15.27	11.12	14.18	9.39	7.28	12.33	9.60
ToG-adapt	14.86	16.64	12.78	14.93	10.71	10.39	14.68	10.59
VoG-adapt	14.97	14.51	12.45	13.39	10.69	9.96	14.70	10.56
KARE-adapt	14.23	12.06	14.10	16.46	12.20	10.68	18.05	8.97
BAR (Qwen3)	17.23	16.20	14.21	**16.56**	11.77	9.01	**18.26**	10.03
BAR	**20.34**	**19.91**	**15.66**	16.54	**13.30**	**10.89**	16.96	**12.39**
MIMIC-IV								
RETAIN	9.22	9.71	6.42	8.62	7.08	4.85	9.08	4.95
StageNet	10.49	10.01	6.88	8.37	7.22	4.95	9.80	5.44
Med-BERT	10.99	11.22	9.08	11.40	7.07	5.23	10.53	6.77
GRAM	11.57	12.60	9.00	11.94	7.15	5.67	10.60	7.73
GraphCare	13.22	14.17	10.77	12.66	10.19	7.27	11.70	10.09
ToG-adapt	14.41	14.05	11.37	13.40	10.61	8.08	14.02	9.45
VoG-adapt	16.76	14.99	11.72	13.51	12.01	8.13	13.86	12.60
KARE-adapt	12.89	16.04	12.55	14.49	11.24	7.78	**18.74**	10.02
BAR (Qwen3)	**19.38**	16.76	**13.51**	16.10	11.96	9.20	15.88	11.49
BAR	19.29	**19.57**	12.55	**17.81**	**12.77**	**10.28**	18.12	**13.06**

Table 14: AUPRC (%) by horizon under the seven-day lead time, against the strongest non-BAR comparator at each horizon. Differences are percentage points.

Data	h	Comparator	Comparator AUPRC	BAR AUPRC	Difference
MIMIC-III	30	VoG-adapt	7.77	8.13	0.36
	90	KARE-adapt	14.09	17.04	2.95
	365	KARE-adapt	18.64	22.07	3.43
MIMIC-IV	30	VoG-adapt	7.79	7.48	−0.31
	90	VoG-adapt	14.08	16.74	2.66
	365	KARE-adapt	18.42	22.07	3.65

D Robustness and Generalization

D.1 Navigation Policy and Candidate Corruption

To isolate the role of the LLM in navigation, we replace plan-guided navigation with two fixed policies at the same assigned budgets. The support-greedy policy expands the candidate with the highest support score, and the random policy picks uniformly among admissible candidates. Support scoring, verification, and the budget rule are unchanged. Both policies lose more AUPRC than removing verification from BAR, and random navigation falls below GraphCare (Table 15). The support-greedy policy cites relevant edges more often than the loop without verification, which suggests that well-attested edges tend to be relevant even when they add little to the prediction.

To test the loop under noisy retrieval, we replace 25% or 50% of the returned candidates at every navigation step on MIMIC-IV with irrelevant edges sampled from G_d, keeping the candidate count and the assigned budget fixed. AUPRC declines linearly with the replaced share, while citation precision declines faster at the higher level and fallback absorbs a growing share of episodes (Table 16). Verification therefore turns many corrupted steps into fallback, but some irrelevant edges still reach the citation list. Even with half of the candidates replaced, BAR stays well above the Stage-1 EHR model that it falls back to.

Table 15: AUPRC and citation precision (Cit.) of navigation policies at matched assigned budgets (%).

	MIMIC-III		MIMIC-IV	
Policy	AUPRC	Cit.	AUPRC	Cit.
BAR	**19.50**	**77.06**	**18.70**	**73.64**
No verification	16.75	61.18	15.99	56.92
Support-greedy	15.79	65.44	15.46	59.96
Random	13.89	59.45	13.01	52.07

Table 16: Candidate replacement on MIMIC-IV (%). Stage-1 EHR is the fallback predictor.

Replaced	AUPRC	Fallback	Cit.
None	18.70	19.75	73.64
25%	17.50	25.90	63.65
50%	16.26	34.09	50.09
Stage-1 EHR	11.17	—	—

D.2 Input Sparsity and Disease Transfer

AUPRC decreases as fewer ICD codes are retained (Figure 6a). BAR maintains the highest AUPRC across the retention levels, and its relative advantage over VoG-adapt grows as codes are removed.

BAR has the highest AUPRC on held-out diseases (Figure 6b). Its relative decline is 7.8%, compared with 6.1% for GraphCare and 10.5% for both Med-BERT and VoG-adapt.

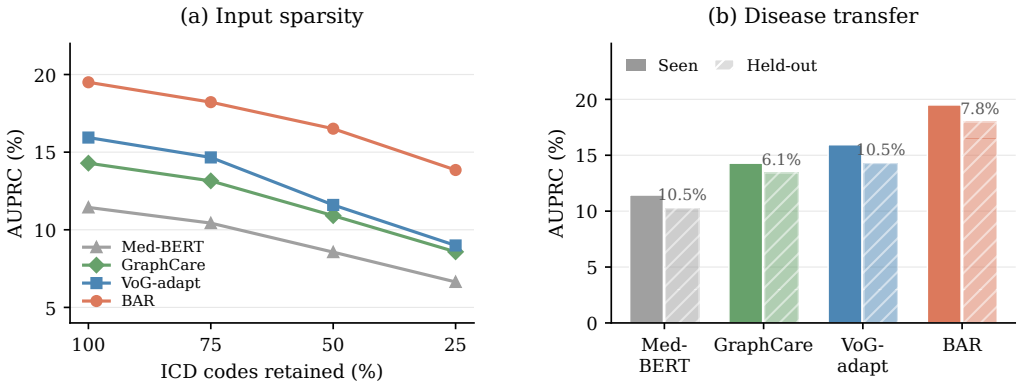

Figure 6: Robustness on MIMIC-III. **(a)** AUPRC when ICD codes are randomly dropped at test time without retraining. **(b)** Leave-two-out disease transfer, where the pipeline is trained on six diseases and only the prediction heads are fine-tuned on the two held-out diseases. Percentages show the AUPRC decline from seen to held-out diseases.

D.3 Knowledge Source and Linker Coverage

To reduce linker coverage, we randomly remove 25% or 50% of the code types that the linker maps, rebuild the anchor sets and evidence graphs, and retrain every KG method. To change the knowledge source, we replace PrimeKG with a graph built from UMLS relations [3] and rerun graph construction with the same settings. Both methods lose AUPRC as linker coverage falls, while BAR remains ahead in every setting (Table 17). The relative margin varies across coverage levels and knowledge sources.

A code type has a unique match when it resolves to exactly one graph node (Appendix B.1). Unmatched rates are highest for external-cause, supplementary, and injury codes, which describe circumstances, encounters, or injuries that seldom have a disease node in the graph (Table 18). Among disease and symptom codes, 62% of MIMIC-III types and 64% and 55% of MIMIC-IV ICD-9-CM and ICD-10-CM types have a unique match. ICD-10-CM adds laterality and encounter detail to many codes, which may explain why fewer of its disease codes resolve to a single node.

Table 17: AUPRC (%) under reduced linker coverage and under UMLS, against the strongest baseline in Table 1.

		MIMIC-III		MIMIC-IV	
Graph	Codes	BAR	ToG	BAR	KARE
PrimeKG	100%	19.50	16.20	18.70	15.88
	75%	17.49	15.18	16.71	14.08
	50%	15.65	13.27	14.90	12.47
UMLS	100%	18.97	15.87	17.88	15.33

Table 18: Code types without a unique graph match, out of all code types. Supplementary codes are ICD-9-CM V or ICD-10-CM Z codes.

	MIMIC-III	MIMIC-IV	
Code group	ICD-9	ICD-9	ICD-10
Disease, symptom	1,446/3,841	1,578/4,390	4,548/10,183
Injury, poisoning	650/838	962/1,219	4,359/4,733
Supplementary	489/579	436/507	1,873/2,067
External causes	231/235	303/306	683/709

D.4 External Site

Among the evaluated methods, BAR achieves the highest AUPRC on five of six EHRSHOT tasks and trails KARE-adapt and Med-BERT on lupus (Table 19). Its macro AUPRC is 13.74, compared with 10.45 for VoG-adapt, the strongest baseline by macro AUPRC. Against KARE-adapt, the paired 95% intervals exclude zero for hyperlipidemia, acute myocardial infarction, and celiac disease. Hyperlipidemia accounts for most of the macro gain over KARE-adapt. Excluding this task, the macro AUPRC is 10.13 for BAR and 9.36 for KARE-adapt, with a difference of 0.78 points calculated before rounding.

Table 19: AUPRC (%) on EHRSHOT. Differences from KARE-adapt and their paired 95% bootstrap intervals are in percentage points. Table 4 gives the test counts.

Task	Med-BERT	VoG-adapt	KARE-adapt	BAR	Difference [95% CI]
Hypertension	16.78	18.42	19.85	**20.50**	0.64 [−1.07, 2.55]
Hyperlipidemia	18.60	26.49	14.85	**31.78**	16.93 [13.51, 20.86]
Acute myocardial infarction	14.00	8.27	14.29	**16.17**	1.88 [0.10, 3.81]
Pancreatic cancer	3.80	5.04	5.52	**6.06**	0.54 [−0.68, 2.20]
Celiac disease	2.53	2.03	2.44	**3.82**	1.37 [0.08, 5.66]
Lupus	4.33	2.45	**4.68**	4.12	−0.56 [−2.81, 3.14]
Macro	10.00	10.45	10.27	**13.74**	—

D.5 Laboratory and Medication Inputs

On MIMIC-IV, we add the last laboratory values recorded before discharge and the medications active at discharge to the patient input of every method, with the same splits and training budget. Every method improves (Table 20). KARE-adapt has the largest gain, and EHR-only methods do not gain consistently more than graph-based ones. BAR keeps the highest AUROC and AUPRC, with a 1.3-point AUPRC margin over KARE-adapt.

Table 20: Results on MIMIC-IV with laboratory and medication inputs. AUROC and AUPRC are percentages, and the gain is the AUPRC change from diagnosis-only inputs in percentage points.

Method	AUROC	AUPRC	AUPRC gain
RETAIN	74.04	11.64	2.51
StageNet	75.75	11.87	2.21
Med-BERT	76.19	12.62	1.58
GRAM	77.58	13.78	1.32
GraphCare	79.10	16.13	2.44
ToG-adapt	79.73	16.89	2.33
VoG-adapt	79.93	17.01	1.60
KARE-adapt	80.02	18.90	3.02
BAR (Qwen3)	80.48	18.48	1.63
BAR	**82.87**	**20.18**	1.48

E Clinical Utility

For each model, we fit a monotone recalibration map on the validation split and apply it unchanged to the test split. The Brier score and the integrated calibration index (ICI) [2] are computed per task on the probability scale and macro-averaged. ICI measures the mean absolute difference between predicted probabilities and a smoothed calibration curve. Net benefit at threshold t is $\mathrm{TP}/n - (\mathrm{FP}/n)\, t/(1-t)$ [35], where n is the number of test observations. We report net benefit per 1,000 observations, macro-averaged over the eight diseases. Positive decisions represent referral for follow-up review. Reviewing every patient is the reference strategy, and reviewing none has zero net benefit. For each horizon, the three thresholds sit below, near, and above the macro-averaged test prevalence.

The raw Stage-1 Brier scores exceed the task-wise constant test-prevalence reference on both datasets. Validation-based recalibration reduces the Brier score and ICI for all three models. BAR has the lowest Brier score before and after recalibration (Table 21). Decision curves use the recalibrated probabilities. All three models achieve higher net benefit than reviewing every patient at the evaluated thresholds, and BAR has the highest net benefit across all 18 settings (Table 22).

Table 21: Brier score and ICI before and after validation-only recalibration, on the probability scale.

		Brier		ICI	
Data	Model	Before	After	Before	After
MIMIC-III	Stage-1 EHR	0.0260	0.0223	0.0271	0.0058
	KARE-adapt	0.0248	0.0217	0.0272	0.0060
	BAR	0.0238	**0.0209**	0.0273	0.0058
MIMIC-IV	Stage-1 EHR	0.0427	0.0366	0.0335	0.0088
	KARE-adapt	0.0397	0.0352	0.0326	0.0088
	BAR	0.0381	**0.0344**	0.0328	0.0089

F Hyperparameter Sensitivity

F.1 Reasoning Hyperparameters

Figure 7 shows how prediction and citation quality change around the default settings. Lowering the support threshold below its default admits weaker evidence and reduces both metrics. Raising it improves citation precision but lowers AUPRC, suggesting that useful evidence is also excluded. The default $\tau_s = 0.3$ balances these effects.

Increasing the pre-filter cap offers little prediction benefit and lowers citation precision. We retain $K_{\mathrm{pre}} = 5$ to keep the candidate list short. At inference, K_C limits the final citation list after prediction. Returning more edges lowers citation precision while leaving AUPRC unchanged. The curves level off once K_C reaches the evidence limit $K_E = 8$.

Table 22: Decision curves after recalibration. Test prevalence π and thresholds t are percentages, and net benefit is per 1,000 test observations. Horizons h are in days.

Data	h	π	t	Stage-1 EHR	KARE-adapt	BAR	Review all
MIMIC-III	30	1.32	0.5	8.29	8.80	**9.29**	8.23
			1.5	4.16	5.33	**6.23**	−1.84
			3.0	2.37	3.48	**4.22**	−17.33
	90	2.40	1.0	15.28	16.14	**16.86**	14.17
			2.5	9.23	10.74	**12.69**	−1.00
			5.0	4.86	6.72	**8.96**	−27.34
	365	3.53	1.0	26.12	26.69	**27.58**	25.52
			4.0	13.49	15.79	**17.74**	−4.93
			8.0	6.53	9.19	**11.89**	−48.62
MIMIC-IV	30	1.80	0.5	13.14	13.22	**13.85**	13.07
			2.0	6.40	7.26	**8.46**	−2.03
			4.0	3.21	4.27	**5.48**	−22.91
	90	3.58	1.0	26.56	27.01	**28.10**	26.06
			3.5	14.28	16.16	**18.91**	0.83
			8.0	6.10	8.50	**11.45**	−48.05
	365	6.75	2.0	50.45	51.82	**53.21**	48.44
			6.5	26.68	32.94	**36.34**	2.64
			12.0	11.07	19.37	**24.05**	−59.69

AUPRC changes little across moderate values of the scorer weight, while citation precision is highest at the default $w_c = 0.4$. Both metrics decline when either the textual or curated signal receives most of the weight, supporting their combined use.

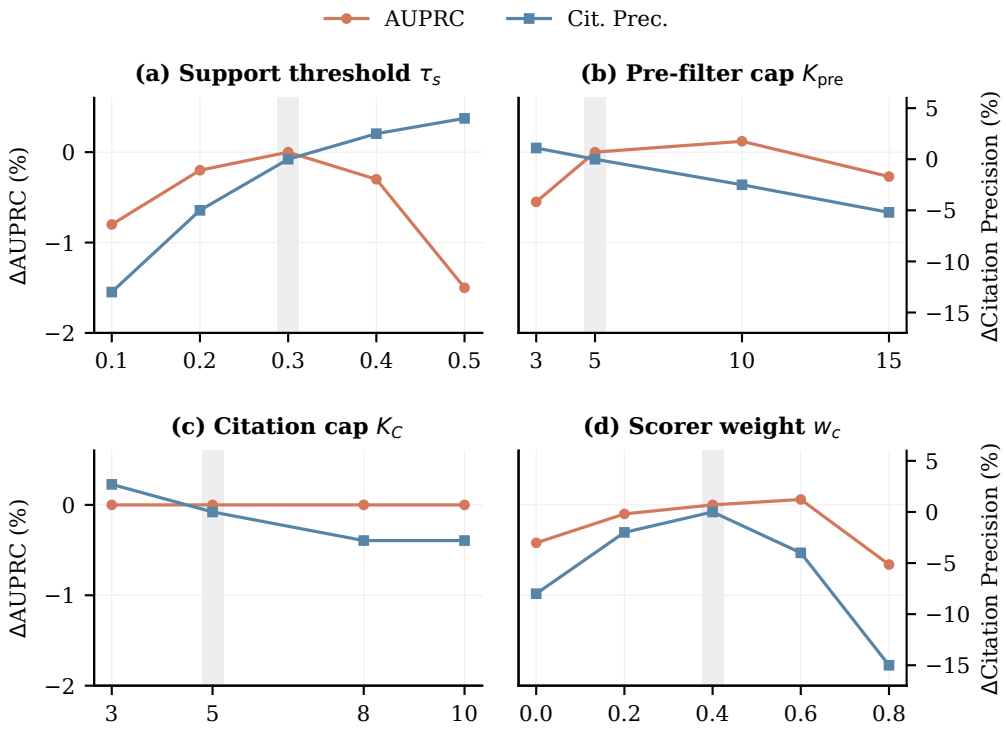

Figure 7: Sensitivity to key hyperparameters, averaged across both datasets. Changes are relative to the default setting in each panel. Gray bands mark defaults.

F.2 Budget Ceiling

Table 23 lists the sweep behind Figure 3 for each dataset. Token use grows quickly up to $B_{\max} = 8$ and slowly afterward, which fits the stopping rule and the evidence cap ending many episodes before the budget is spent, as the 62–65% utilization at the default also shows.

Table 23: Budget sweep. Tokens are mean LLM tokens per episode, and AUPRC is a percentage. The default is $B_{\max} = 12$.

$B_{\max}$		2	4	6	8	10	12	14	16	18	20
MIMIC-III	Tokens	3,677	5,674	7,484	8,957	9,857	10,276	10,673	11,047	11,373	11,655
	AUPRC	15.78	18.67	19.68	**19.72**	19.62	19.50	19.38	19.28	19.18	19.10
MIMIC-IV	Tokens	3,916	6,043	8,015	9,696	10,806	11,325	11,691	12,028	12,308	12,547
	AUPRC	14.87	17.56	18.62	**18.78**	18.77	18.70	18.62	18.54	18.47	18.41

F.3 Graph Construction

We vary K, $d_{\max}$, M, and anchor frequency one at a time. For each dataset, the default graph and four alternatives start independently from each seed's default checkpoint. Each receives five additional stage 3 sampling rounds of up to 4,096 episodes, with matched patient–disease–horizon draws across settings. Routing and supervised updates follow Appendix B.3. Embeddings are copied by node and relation identifier, and newly introduced embeddings use Xavier initialization. Training samples all three horizons, while validation and testing for this analysis use 365 days. Each setting retains its fifth-round checkpoint.

On MIMIC-IV, one-shot retrieval, support-greedy navigation, removal of verification, and removal of plan revision are applied at inference to each setting's checkpoint. Each variant reruns the reasoning loop with model parameters fixed. Drops are measured against the full model from the same checkpoint.

Full-model AUPRC stays within 0.6 points of the default and above the strongest baselines at this horizon, ToG-adapt on MIMIC-III (20.87) and KARE-adapt on MIMIC-IV (21.16). The ablations keep their ranking except at $M = 20$, where support-greedy navigation costs slightly more than one-shot retrieval.

Table 24: Graph-construction sensitivity at 365 days after five rounds of adaptation. Default denotes the adapted default graph with $K = 3$, $d_{\max} = 50$, $M = 10$, and anchor frequency 1%. Full-model AUPRC is a percentage. MIMIC-IV ablation drops are percentage-point differences from the same row's full model, with model parameters fixed during ablation.

	Full model		AUPRC drop on MIMIC-IV			
Setting	MIMIC-III	MIMIC-IV	One-shot	Support-greedy	w/o verification	w/o revision
Default	24.91	24.35	4.52	3.68	3.09	1.43
$K = 2$	24.38	24.09	3.76	3.34	2.63	1.12
$d_{\max} = 100$	24.72	23.98	4.41	3.95	3.42	1.59
$M = 20$	25.06	24.58	4.18	4.29	3.17	1.36
Anchor freq. 0.5%	24.67	24.22	4.67	3.52	2.88	1.54

G Evidence-Quality Score Analysis

Table 25 groups observations by median support and patient relevance. The Pearson correlation between the scores is 0.228. Selection is highest when both scores are high, lowest when both are low, and similar in the two mixed groups.

Table 25: Selection rates (%) and selected/total counts for 40,000 observations, split at the median of each score. Marginal rates pool the counts.

	High Relevance	Low Relevance	Marginal Rate
High Support	66.5 (7,699/11,584)	30.5 (2,571/8,416)	51.4
Low Support	31.1 (2,620/8,416)	16.5 (1,913/11,584)	22.7
Marginal Rate	51.6	22.4	—

H Citation Precision Evaluation

Citation precision measures the relevance of returned edges to the patient's risk pathway. For each method, dataset, seed, and evaluated setting, we sample up to 5,000 returned citation occurrences uniformly without replacement. All occurrences are used when fewer are available. Sampling treats repeated citations from different episodes as separate occurrences. GPT-4o [16] receives the discharge codes, target disease, and cited edge with its support score, and returns a binary relevance label with a short explanation at temperature 0 [43]. The prediction and the edge's position in the reasoning trace are hidden. Citation precision is the proportion labeled relevant in the sample. Judgments are reused for identical complete judge inputs. The 300 doubly annotated pairs below assess judge agreement separately.

Annotation protocol. The comparison covers 300 patient-edge pairs. Both annotators label the same pairs independently using the judge's inputs and relevance criterion.

Agreement. Table 26 reports pairwise agreement and judge performance on the 254 pairs with a shared annotator label. Appendix I gives the judge prompt.

Table 26: Pairwise Cohen's κ on 300 patient-edge pairs. The lower panel evaluates the judge on the 254 pairs where both annotators agree. Precision and recall refer to the relevant class.

	Annotator A	Annotator B	GPT-4o
Annotator A	—	0.69	0.76
Annotator B	0.69	—	0.65
GPT-4o	0.76	0.65	—
Judge vs. shared label	κ	Precision	Recall
Consensus subset	0.83	0.89	0.93

I LLM Prompt Templates

We use five prompt types, one for plan generation, one for edge selection during navigation, one for verification, one for plan revision, and one for the citation judge. All reasoning prompts receive the patient's discharge codes X_i, history H_i, target disease d, horizon h, and remaining budget.

Plan generation

Instruction: Decompose a clinical risk assessment into evidence-gathering steps over a medical knowledge graph. Each step traces a specific clinical pathway (comorbidity cascade, drug interaction, or disease progression) from the patient's conditions toward the target disease. Allocate budget fractions that sum to at most 1.0.

Inputs

Patient codes	$\{X_i\}$	History	$\{H_i\}$
Target disease	d	Horizon	h days
Anchor set	$A_{i,d}$	Budget	B

Output: $P = [s_1, \ldots, s_n]$ where $s_t = (\text{hypothesis } h_t,\ \text{query } q_t,\ \text{expected } \hat{f}_t,\ \text{budget } b_t)$.

Example s_1 : hypothesis "AF raises clot risk", query "AF $\rightarrow$ thromboembolism", expected "stasis, hypercoagulability", budget 0.35

Edge selection (navigation)

Instruction: Given the current hypothesis and a ranked list of candidate edges that have already been filtered by support threshold and deduplicated, select which edges to retain as evidence for this step. Prioritize edges that directly support the hypothesis.

Inputs

Current step	$s_t = (h_t, q_t, \hat{f}_t, b_t)$
Candidates	$\{(u_j, r_j, v_j, \phi_t(e_j))\}_{j=1}^{K_{\text{pre}}}$, ranked by ϕ_t
Remaining capacity	$K_E - \lvert E \rvert$

Output: $E_t \subseteq$ candidates, with at most $K_E - |E|$ selected edge IDs.

Verification

Instruction: Verify whether the retrieved evidence supports the plan step hypothesis. A step passes if the edges are clinically consistent with the hypothesis and the average quality score is sufficient. A step fails if the evidence is absent, contradictory, or too weak to support the hypothesis.

Inputs

Hypothesis	h_t	Expected	$\hat{f}_t$
Retrieved	E_t with $\{\phi_t(e)\}$		

Output: `PASS` | `FAIL` with one-sentence reason.

Plan revision

Instruction: Regenerate the plan from the failed step onward using a different clinical pathway that avoids the failure mode. Previously verified steps are frozen. The revised plan must fit within the remaining budget.

Inputs

Failed step	s_t with E_t and failure reason
Verified context	$\{(s_j, E_j)\}_{j=1}^{t-1}$
Remaining budget	$B - C_t$

Output: A revised plan for the remaining steps.

Citation relevance judge

Instruction: Determine whether a knowledge graph edge cited by a clinical reasoning system is relevant to the patient's risk for the target disease. An edge is relevant if it describes a mechanism, risk factor, or pathway connecting the patient's conditions to the target.

Inputs

Patient codes	$\{X_i\}$
Target disease	d
Cited edge	(u, r, v) with support $s(e)$

Output: `RELEVANT` | `NOT_RELEVANT`, with one-sentence justification.

J Broader Impact and Ethics

Data use. The study design uses de-identified MIMIC-III [22] and MIMIC-IV [21] records under the PhysioNet Credentialed Health Data License and de-identified EHRSHOT [37] records under its research data use agreement. UMLS is used under the UMLS Metathesaurus License.

Intended use and limitations. BAR is a research prototype for budget-aware reasoning over medical knowledge graphs. Clinical deployment would require prospective validation and expert review of its reasoning traces, neither of which is included in this study.

Potential risks. PrimeKG [5] may be incomplete or outdated, and LLM reasoning may be clinically incorrect. Follow-up covers only admissions within each health system, with no recorded readmission labeled negative, so diagnoses made elsewhere are unobserved. Fixed-horizon labels do not model censoring or competing risks such as death. Raw risk scores are miscalibrated and need local recalibration before use as probabilities. Citation evaluation checks whether cited edges are relevant

to the patient, and the medical correctness of full reasoning traces remains untested. Generalization to other populations and subgroup differences by age, sex, race, or insurance remain untested.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the evidence graphs, budget-aware reasoning loop, and training objective. Their empirical claims follow Table 1, Figure 3, Figure 5, and Appendix D.4, within the scope stated in Section 3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 states that labels cover only diagnoses recorded within one health system and do not model censoring or competing risks, that raw scores need local recalibration, that the medical correctness of full reasoning traces is untested, and that prospective clinical validation is still needed. Section 3.4 reports that the margin over the baselines disappears at 30 days under a seven-day lead time. Tables 7 and 8 report the LLM calls and tokens per episode, and Appendix J discusses further risks, including untested subgroup differences.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theoretical results or proofs. The contributions are methodological and empirical.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We specify datasets, target diseases, prediction horizons, and a patient-disjoint split based on recorded dates. The methodology and appendix describe the EHR encoder, support scorer, hyperparameters, training procedure, reasoning-loop pseudocode, baseline adaptations, and the lead-time and EHRSHOT protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: MIMIC-III and MIMIC-IV are available through PhysioNet credentialed access, EHRSHOT under its research data use agreement, and UMLS under its license, and PrimeKG is publicly available. Code is released at the repository linked in Section 1. The paper provides sufficient algorithmic detail for independent reimplementation.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experiment section describes datasets, baselines, and metrics. The appendix gives the model configuration, patient-disjoint recorded-date split, baseline adaptations, and the details of cohort extraction, linking, and policy optimization.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports means and sample SDs over three independent training seeds. Table 19 reports paired patient-bootstrap intervals. Appendix B.4 specifies the run allocation, and Appendix H describes citation sampling.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Table 6 reports hardware, and Tables 7 and 8 report LLM calls and tokens per test episode. Training times and total compute are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses only de-identified records accessed under their data use terms, namely MIMIC-III and MIMIC-IV under the PhysioNet Credentialed Health Data License and EHRSHOT under its research data use agreement. No re-identification was attempted. Appendix J discusses data use, intended use, limitations, and potential risks.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix J discusses potential risks, including outdated KG knowledge, clinically plausible but incorrect LLM reasoning, miscalibrated raw scores, unobserved diagnoses outside the health system, and untested subgroup differences, and it states that clinical use would require prospective validation and expert review of the reasoning traces.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pre-trained models or scraped datasets. BAR is a research prototype evaluated on existing public benchmarks and poses no direct risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: MIMIC-III and MIMIC-IV are cited and used under the PhysioNet Credentialed Health Data License, EHRSHOT under its research data use agreement, and UMLS under the UMLS Metathesaurus License. PrimeKG is cited. The LLMs (GPT-4o-mini, GPT-4o, Qwen3-8B) are cited with their respective references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper releases its code at the repository linked in Section 1.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. The two-annotator validation in Appendix H (300 instances for citation precision) is an internal evaluation step, not a human-subjects study.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper uses only de-identified records (MIMIC-III, MIMIC-IV, and EHRSHOT) under their existing data use agreements. No human subjects research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs (GPT-4o-mini, Qwen3-8B) are core components of the proposed method, serving as the reasoning agent in the budget-aware reasoning loop. Their role is described throughout the Methodology section, with prompt templates in Appendix I. GPT-4o serves as the citation judge (Appendix H).

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.