EHRBench: An Automated and Reliable EHR-based Benchmark for Clinical Decision Making with LLMs

Yuzhang Xie
Emory University
Atlanta, GA, USA
yuzhang.xie@emory.edu

Keqi Han
Emory University
Atlanta, GA, USA
keqi.han@emory.edu

Yunpeng Xiao
Emory University
Atlanta, GA, USA
yunpeng.xiao@emory.edu

Hejie Cui
Stanford University
Stanford, CA, USA
hejie.cui@stanford.edu

Guanchen Wu
Emory University
Atlanta, GA, USA
guanchen.wu@emory.edu

Ziyang Zhang
Emory University
Atlanta, GA, USA
ziyang.zhang2@emory.edu

Kai Shu
Emory University
Atlanta, GA, USA
kai.shu@emory.edu

Jiaying Lu
Emory University
Atlanta, GA, USA
jiaying.lu@emory.edu

Xiao Hu
Emory University
Atlanta, GA, USA
xiao.hu@emory.edu

Carl Yang*
Emory University
Atlanta, GA, USA
j.carlyang@emory.edu

Abstract

Clinical decision-making (CDM) is central to real-world clinical workflows, where clinicians infer diagnoses, select treatments, or anticipate future health outcomes under incomplete evidence. LLMs are increasingly used to support these decisions due to strong language capabilities, broad biomedical knowledge, and efficiency, yet the reliability of LLMs on real-world clinical decision tasks remains insufficiently understood. To evaluate CDM models, especially LLM-based models, an ideal and practical medical decision benchmark should be constructed via an automated yet reliable pipeline to ensure both scale and quality. Moreover, the grounding of a CDM benchmark in real patient EHRs can better support evaluation on practical CDM tasks that require substantive biomedical knowledge and clinical inference. To fill the gaps, we introduce EHRBench, an automated and reliable EHR-grounded benchmark for evaluating LLM-based clinical decision-making at scale. To ensure scalability and reliability, EHRBench is constructed through an EHR–LLM–knowledge-base (KB) interaction pipeline. For efficiency, we use a specialized LLM to automatically convert encounter-level EHR trajectories into structured templates and deterministically instantiate the templates into QA items. In parallel, we apply systematic KB-based verification and enrichment to filter hallucinated or ambiguous relations and to improve reliability. Using this pipeline, we construct nearly 1M (960,067) QA items spanning three core inference-required clinical decision tasks: diagnosis, treatment, and prognosis. We benchmark more than 30 representative LLMs on EHRBench and provide detailed analyses of performance and robustness. The results show consistent capability trends across settings, further validating the reliability of EHRBench and highlighting actionable gaps toward clinically reliable LLM systems [1].

CCS Concepts

• **Applied computing** → **Health care information systems**; • **Information systems** → **Data mining**; • **Computing methodologies** → **Natural language processing**; **Artificial intelligence**.

Keywords

Large language models; Electronic health records; Clinical decision making; Medical question answering; Benchmark; Knowledge base verification

ACM Reference Format:
Yuzhang Xie, Keqi Han, Yunpeng Xiao, Hejie Cui, Guanchen Wu, Ziyang Zhang, Kai Shu, Jiaying Lu, Xiao Hu, and Carl Yang. 2026. EHRBench: An Automated and Reliable EHR-based Benchmark for Clinical Decision Making with LLMs. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD 2026), August 9–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 38 pages. https://doi.org/10.1145/3770855.3817571

*Carl Yang is the corresponding author.

KDD 2026, Jeju Island, Republic of Korea.

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817571

[1]The guideline for source code and data of EHRBench is available at the GitHub link https://github.com/constantjxyz/EHRBench

1 Introduction

Clinical decision-making (CDM) is a fundamental part of real-world clinical workflows, where clinicians must infer diagnoses, determine treatments, or forecast future clinical states from incomplete evidence [30, 62, 73, 83]. For instance, given the observed diagnoses of an encounter, an *in-encounter diagnosis completion* decision requires inferring concurrent conditions, such as identifying chronic kidney disease when type 2 diabetes and diabetic nephropathy are present. Similarly, an *in-encounter treatment selection* decision involves selecting appropriate treatments, such as identifying the necessary anticoagulation for a patient with atrial fibrillation. Furthermore, a *next-encounter prognosis prediction* decision requires anticipating potential downstream outcomes or diagnoses in subsequent encounters, such as forecasting ischemic stroke risk in patients with hypertension and hyperlipidemia. These decisions directly affect patient care and outcomes, carrying substantial clinical significance for patient safety and well-being [71, 86].

Large language models (LLMs) are increasingly deployed to support these clinical decisions [34, 35, 64, 68, 102, 124], owing to their robust language understanding capabilities, broad biomedical knowledge acquired during pre-training, and superior efficiency relative to traditional manual workflows [9, 46, 60, 82, 89]. This rapid progress raises a central question: how reliably do LLMs perform on core clinical decision tasks when the evidence reflects patient-specific, real-world clinical data? Benchmarks are essential for addressing this question, as they enable controlled, reproducible comparisons and provide guidance for the development of safer CDM systems.

Building these benchmarks requires an automated and reliable construction pipeline. Historically, many medical QA resources have achieved high quality through substantial domain expertise and meticulous manual curation [61, 82, 97, 122]. However, the high cost associated with manual effort typically limits these benchmarks to a small number of patient records, which restricts the scale and diversity of evaluation [11, 110]. Since CDM is inherently complex and multi-faceted, large-scale benchmarks are essential for comprehensive evaluation, which in turn necessitates the transition toward automated construction pipelines.

Recent studies have explored the use of LLMs themselves to scale up benchmark creation by generating questions under specific constraints [5, 59, 80]. While this produces a large volume of data, it raises quality issues since LLMs can hallucinate. Consequently, ensuring that LLM-generated benchmarks are clinically realistic and unambiguous requires more than formatting constraints alone; it necessitates systematic validation (e.g., via external knowledge bases) to mitigate hallucinated clinical relations and ambiguous answers [32, 65].

Beyond constructing an automated and reliable pipeline, the data source of the benchmark is also important. Grounding a CDM benchmark in patients' real electronic health records (EHRs) facilitates more authentic evaluations of practical CDM tasks. Currently, most existing medical benchmarks are derived from well-formed narrative sources such as exams, textbooks, clinical guidelines, and clinical notes [17, 37, 44, 47, 54, 56, 63, 69, 94, 118, 126]. These sources often make clinical reasoning explicit—for example, by directly stating the rationale for diagnoses or treatments—thereby reducing the need for inference. In contrast, clinicians routinely reason over longitudinal EHRs, where the underlying clinical logic is not pre-digested but must be inferred from patterns of structured events. Unlike general medical sources that emphasize idealized and broadly applicable knowledge, EHRs capture personalized, longitudinal, real-world clinical events and care patterns at scale [45, 108, 121]. Furthermore, compared with free-text clinical notes, which are costly to curate and typically focus on a limited set of salient details, the structured tabular components of EHRs have far higher volume, cover a broader range of clinical concepts, and reflect substantially greater variability in real-world practice [43].

Despite this potential, directly leveraging raw structured EHRs for benchmark construction remains challenging. Clinical relationships in EHRs are largely implicit and must be inferred from temporally ordered events, while fragmentation across coding systems complicates faithful transformation into natural-language prompts without introducing artifacts or label leakage [99, 104]. In addition, EHR trajectories are often extremely long, making it difficult to convert raw records into LLM-feasible inputs while preserving data fidelity [78, 120]. As a result, existing EHR-based benchmarks often emphasize reading-comprehension or information-retrieval tasks (e.g., "what treatment did the patient receive during this visit" [49, 109]), rather than core CDM tasks that require substantive biomedical knowledge and clinical inference, such as deciding what should be prescribed given a diagnosis.

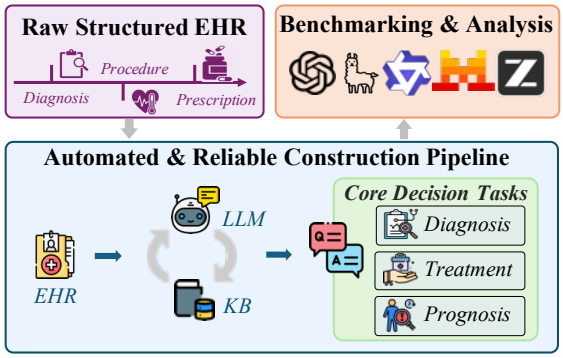

Figure 1: Overview of EHRBench. EHRBench automatically and reliably transforms raw structured EHR trajectories into QA benchmarks via an EHR-LLM-KB interaction pipeline and evaluates representative LLMs on three core clinical decision tasks: diagnosis, treatment, and prognosis.

To bridge these gaps, we introduce EHRBench, an automated and reliable benchmark grounded in real-world electronic health records (EHRs) for evaluating the clinical decision-making capabilities of LLMs. As illustrated in Figure 1, our framework systematically transforms raw structured EHR trajectories into a benchmark that is both large-scale and high-quality through a multi-stage pipeline that integrates EHR data, LLMs, and external biomedical knowledge bases (KBs). Specifically, we use LLMs to generate question templates (including clinical relations, questions, and answers) from EHR trajectories, which are concurrently validated (for clinical relations) and enriched (with entity definitions and retrieved evidence) using external KBs to ensure clinical reliability. These

generated templates are deterministically instantiated into multiple types of QA items to ensure diversity. Using EHRBench, we evaluate representative LLMs on three core CDM tasks that require substantive biomedical knowledge and clinical inference, covering in-encounter diagnosis completion (*diagnosis*), in-encounter treatment selection (*treatment*), and next-encounter outcome prediction (*prognosis*). We further analyze model performance in terms of accuracy, efficiency, and robustness.

Our contributions are summarized as follows:

- We construct EHRBench, a large-scale, EHR-grounded QA benchmark for evaluating LLMs' clinical decision-making capabilities, comprising nearly 1 million QA items (960,067). To the best of our knowledge, EHRBench is the first benchmark built directly from raw structured EHR trajectories that leverages LLMs for question template generation while enforcing systematic verification for clinical reliability.
- We propose an automated and reliable benchmark construction framework based on EHR–LLM–KB interactions, where LLMs enable scalable template generation, KBs provide principled validation and enrichment, and EHR trajectories supply realistic longitudinal clinical evidence.
- We formulate clinical decision making as conditional inference over partially observed EHR data and design three representative tasks: diagnosis completion, treatment selection, and next-encounter prognosis, which require substantive biomedical knowledge and clinical inference over implicit clinical relations and longitudinal patient trajectories.
- We systematically benchmark more than 30 representative LLMs on EHRBench and conduct comprehensive analyses of their accuracy, efficiency, and robustness, providing actionable insights for developing and evaluating clinically reliable LLM systems.

2 Related Work

Medical QA Benchmarks. Medical QA benchmarks are essential for measuring the biomedical knowledge and reasoning capabilities of clinical decision-supporting models, including LLMs [101]. A large body of work constructs high-quality QA resources through expert curation or carefully designed evaluation protocols. These approaches typically improve correctness and reduce ambiguity, but they often limit dataset scale due to annotation cost and the need for domain expertise, including MedAlign [23], SD-Bench [66], ExpertQA [61], and MedThink-Bench [122], which typically contain several hundred expert-annotated QA pairs. Most existing large-scale medical benchmarks are derived from general narrative sources such as exams, textbooks, and clinical guidelines, including MedQA [37], MedMCQA [69], ClinicBench [54], MedXpertQA [126], MedChain [56], MedExQA [44], LLM-Eval-Med [118], TrialPanorama [94], CHBench [27], CMB [91], MedOdyssey [21], MedSBench [98], MultiFacetEval [123], ReasonMed [84], XMedBench [92], and related evaluations of medical reasoning and generalization. In addition, several benchmarks are grounded in healthcare practice-generated clinical notes, case reports, or dialogue-style clinical interactions, such as MediSumQA [17], EHRNoteQA [47], ER-REASON [63], CPUCase [74], LongHealth [2], MedR-Bench [75], MMMU [116], HealthBench [4], DiagnosisArena [125], and CRAFT-MD [40]. Safety-centered medical benchmarks further evaluate risk, harmfulness, and reliability in clinical contexts, such as MedSafetyBench [29] and MedRisk or related risk-oriented agents [55]. Beyond general QA, some benchmarks focus on specialized competencies such as medical calculation [41], concept-centric QA [79], or epidemiological question answering [95]. A separate but related direction builds agentic or interactive environments for sequential diagnosis and decision support, including MEDIQ [51], AI Hospital [22], AgentClinic [77], MAQUE [25], VivaBench [14], AgentHospital [50], MMD-Eval [58], and AMIE [85]. Moreover, multimodal information, including ECG, genomics, imaging, and other medical data, is becoming increasingly important for CDM by providing complementary evidence about patient physiology and disease status [28, 90, 93, 106, 107]. This trend has motivated multimodal medical benchmarks such as Asclepius [57], CLIMB [18], EHRXQA [6], GMAI-MMBench [114], OmniMedVQA [31], and PMC-VQA [119]. Despite this breadth, few benchmarks are built directly from raw structured EHR trajectories in a way that preserves real-world patterns required for CDM.

EHR QA Benchmarks. A growing body of work leverages raw EHR data to construct QA datasets and benchmarks [8]. However, most existing resources primarily assess the ability of a model to retrieve explicit facts from large, redundant tabular records, rather than to infer clinical decisions from longitudinal context [103]. A representative line of work frames EHR QA as text-to-SQL parsing or database querying, including benchmarks such as EHRSQL [49] and emrQA [70], as well as systems that map questions to executable queries (e.g., emrKBQA [76], MIMICSQL [88]) or agentic coding workflows [109]. Complementary knowledge-graph-based approaches, such as ClinicalKBQA [87] and MIMIC-SPARQL [72], and temporal-reasoning benchmarks like TIMER [16], further enable relational and time-aware querying. Finally, concurrent efforts increasingly evaluate LLMs on clinical decision tasks (e.g., EHR-R1 [52]), underscoring the importance and urgency of our work. In contrast to these efforts, prior work does not emphasize an automated and reliable EHR-LLM-KB pipeline that explicitly extracts clinical relations from raw structured EHRs and then systematically verifies and filters them using large-scale knowledge bases.

Positioning of Our Work. Complementary to prior benchmarks that rely on curated narratives or emphasize retrieval-oriented EHR QA, our work targets realistic CDM evaluation by (i) grounding the benchmark in raw structured EHR trajectories, (ii) formulating three core CDM tasks that require substantive biomedical knowledge and clinical inference beyond information access, (iii) using LLMs to extract implicit clinical logic from raw EHR data for efficiency, and (iv) enforcing systematic verification and enrichment via biomedical KBs to maintain reliability.

3 EHRBench Construction Methodology

3.1 Problem Definition & Framework

Our goal is to transform structured EHRs into a clinically grounded QA benchmark for evaluating LLMs through an automated and reliable pipeline. Figure 2 presents an overview of the construction framework. The pipeline first preprocesses raw structured EHRs and normalizes clinical events into a standardized encounter-level representation. It then constructs templates through an automated

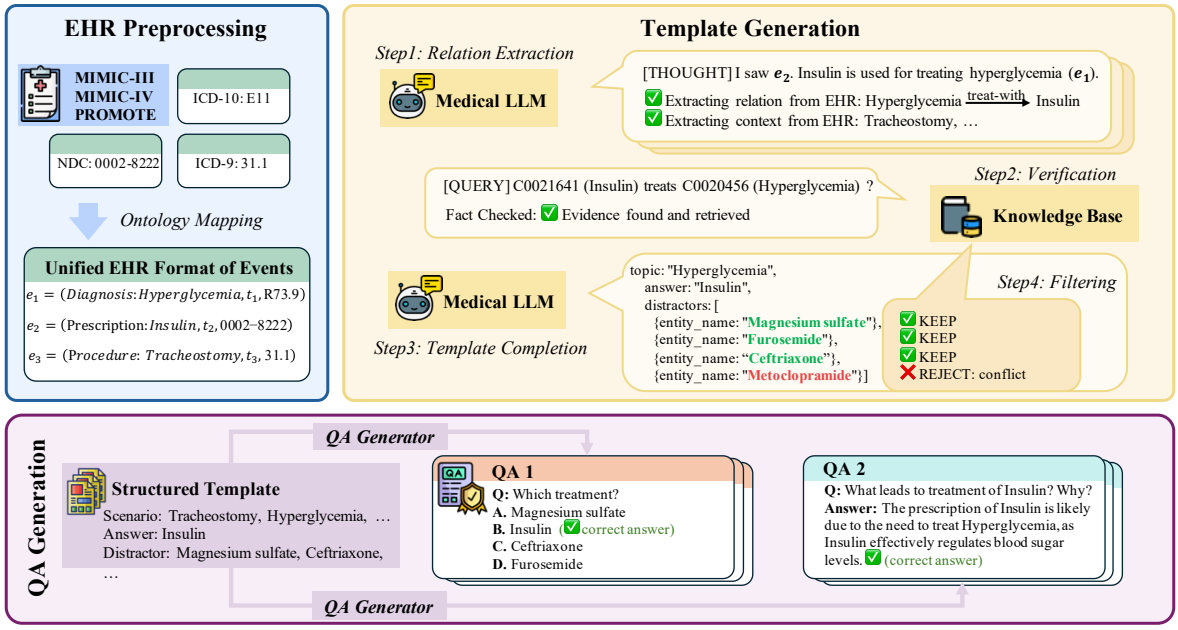

Figure 2: Construction pipeline of EHRBench. Starting from raw structured EHRs, we preprocess and normalize encounter-level clinical events into a standardized representation. We then generate structured templates integrating EHR signals, LLM-based extraction, and KB verification and enrichment. Finally, the QA generation module deterministically instantiates each template into multiple QA items for downstream evaluation. Overall, the pipeline is LLM-driven for scalability, KB-verified for reliability, and EHR-grounded for clinical relevance.

EHR-LLM-KB interaction pipeline that extracts clinically meaningful signals and validates and enriches them through KB evidence. Finally, the pipeline instantiates each template into multiple QA variants and question formats, yielding task-specific QA items for evaluation. In summary, the pipeline is LLM-powered for scale, KB-checked for reliability, and grounded in real EHR data for clinical relevance. We describe each step in detail below.

EHR data collection & representation. Let

$$\mathcal{E} = \{\mathcal{E}^{(1)}, \mathcal{E}^{(2)}, \ldots, \mathcal{E}^{(N)}\} \tag{1}$$

denote a cohort of EHRs from N encounters, where $\mathcal{E}^{(n)}$ is the structured record associated with encounter n. We assume each encounter $\mathcal{E}^{(n)}$ is associated with a patient identifier $\pi(n)$, and encounters are ordered chronologically within each patient.

Each encounter-level EHR $\mathcal{E}^{(n)}$ is represented as a set of recorded clinical events:

$$\mathcal{E}^{(n)} = \left\{e_1^{(n)}, e_2^{(n)}, \ldots, e_{M_n}^{(n)}\right\}, \tag{2}$$

where M_n is the number of events observed in encounter n.

Each event $e_m^{(n)}$ is represented as

$$e_m^{(n)} = \left(d_m^{(n)}, t_m^{(n)}, a_m^{(n)}\right), \tag{3}$$

where $d_m^{(n)}$ is a textual description of the clinical event (e.g., diagnosis or prescription), $t_m^{(n)}$ is a timestamp, and $a_m^{(n)}$ denotes additional attributes such as medical codes or numerical values.

In practice, we follow common conventions by treating encounters as the basic temporal unit, aggregating clinical events at the encounter level rather than collapsing them to the patient level or relying on fine-grained timestamps. Purely patient-level aggregation is overly coarse because it ignores temporal structure and distinctions across encounters, while fine-grained timestamps are fragmented and reflect administrative logging rather than clinical onset (e.g., many diagnosis events are summarized as billing codes at discharge). Using encounters as the primary temporal unit aligns representations with documentation and CDM, enabling consistent aggregation over meaningful windows while allowing downstream tasks to focus on within-encounter evidence or longitudinal history.

Additional details regarding data sources and cohort construction are provided in Section 3.2.

Template generation. A template generation function is

$$g : \mathcal{E} \to \mathcal{P}, \tag{4}$$

which maps the encounter cohort $\mathcal{E}$ to K structured QA templates:

$$\mathcal{P} = \{P_k\}_{k=1}^{K}. \tag{5}$$

The construction of $\mathcal{P}$ is a multi-stage interaction among EHR data, LLMs, and biomedical knowledge bases (KBs). Each template P_k defines a clinically grounded blueprint that can be deterministically instantiated into one or more QA items. Each P_k comprises:

- A template context $C_k \subset \mathcal{E}^{(n)}$, constructed by an LLM by selecting relevant events from an encounter record $\mathcal{E}^{(n)}$.
- A clinical relation $R_k = (x_k, r_k, y_k)$, where x_k and y_k are the subject and object entities and r_k is the relation predicate that links them, such as (*Hypertension*, *Cause*, *Stroke*). Each

relation is extracted from EHR events using an LLM and verified against KBs to ensure clinical validity.
- A set of latent attributes A_k, generated by the LLM or retrieved from KBs, including entity definitions, evidence or rationale, candidate distractors, and the clinical topic associated with the relation.

All attributes in P_k are exposed to the LLM during the subsequent QA generation stage to provide guidance. Additional details of the template generation procedure are provided in Section 3.3.

QA generation. A transformation function is defined as

$$f : \mathcal{P} \to \mathcal{I}, \quad \text{where } \mathcal{I} = \left\{(S_j, Q_j, B_j)\right\}_{j=1}^{J}, \tag{6}$$

which maps the constructed templates to a collection of J constructed QA items. Each QA item consists of:
- a textual scenario S_j, which is a natural-language paragraph that verbalizes some background clinical events from a patient encounter;
- a natural-language question Q_j constructed from the template,
- a metadata bundle B_j, including the choices, correct answer, clinical rationale, associated medical topic, and underlying clinical relations.

More details of the QA generation are provided in Section C.4.

Clinical decision tasks. Using the formulation above, a collection of QA items $\mathcal{I}$ is constructed to target three core clinical decision tasks that require medical knowledge and inference. Each task corresponds to a conditional inference objective grounded in encounter-level EHR data $\mathcal{E}$, where the model receives a scenario $S^{(n)}$ composed of a subset of observed events.

(I) Diagnosis decision (in-encounter diagnosis completion). This task evaluates intra-encounter diagnostic inference by predicting a missing diagnosis from other diagnoses recorded in the same encounter (referred to as "diagnosis decision" in this study for brevity). Given an encounter n with diagnosis set $\mathcal{D}^{(n)}$, we withhold a target diagnosis $d_{\text{tgt}}^{(n)} \in \mathcal{D}^{(n)}$ and create a scenario diagnosis subset

$$\mathcal{S}^{(n)} \subseteq \mathcal{D}^{(n)} \setminus \left\{d_{\text{tgt}}^{(n)}\right\}. \tag{7}$$

The model is asked to infer the missing diagnosis:

$$d_{\text{tgt}}^{(n)} \sim p\left(d \,\middle|\, \mathcal{S}^{(n)}\right). \tag{8}$$

Accordingly, the scenario description S_j verbalizes $\mathcal{S}^{(n)}$, and the question asks for the most likely co-occurring diagnosis. This task measures whether the model captures clinically plausible comorbidity patterns and diagnostic co-occurrence structure within a single encounter.

(II) Treatment decision (in-encounter treatment selection). This task models encounter-level treatment selection (referred to as "treatment decision" in this study for brevity). Given encounter n, a scenario diagnosis set is constructed as

$$\mathcal{S}^{(n)} \subseteq \mathcal{D}^{(n)} \tag{9}$$

and the model is required to infer a target treatment $t_{\text{tgt}}^{(n)}$ prescribed or performed during the same encounter:

$$t_{\text{tgt}}^{(n)} \sim p\left(t \,\middle|\, \mathcal{S}^{(n)}\right). \tag{10}$$

Here, the scenario description S_j verbalizes $\mathcal{S}^{(n)}$, and the question asks the model to select an appropriate treatment from $\mathcal{T}^{(n)}$ (i.e., a prescription or a procedure).

(III) Prognosis decision (next-encounter outcome prediction). This task evaluates longitudinal reasoning over consecutive encounters to anticipate future diagnoses (referred to as "prognosis decision" in this study for brevity). Given two consecutive encounters n and n+1 for the same patient, a scenario event set is constructed as

$$\mathcal{S}^{(n)} \subseteq \mathcal{D}^{(n)} \cup \mathcal{T}^{(n)} \tag{11}$$

from encounter n, and the model is required to predict a target diagnosis $d_{\text{tgt}}^{(n+1)}$ in the subsequent encounter:

$$d_{\text{tgt}}^{(n+1)} \sim p\left(d \,\middle|\, \mathcal{S}^{(n)}\right). \tag{12}$$

Here $\mathcal{D}^{(n)}$ and $\mathcal{T}^{(n)}$ denote the diagnoses and treatments (including procedures and prescriptions) observed at encounter n. For each QA item j, the scenario description S_j is a natural-language rendering of $\mathcal{S}^{(n)}$, and the question asks for a diagnosis that appears at encounter n+1. This task evaluates whether disease progression and treatment-related effects over time under partial observation are captured.

Overall, the resulting benchmark is designed to systematically evaluate LLMs' ability to perform clinically grounded reasoning and decision-making over structured, longitudinal EHR data under partial observation. Details of the evaluation protocol are summarized in Appendix C.7.

3.2 Data Collection & Preprocessing

Our benchmark utilizes structured EHR trajectories from three real-world sources: MIMIC-III, MIMIC-IV, and PROMOTE. MIMIC-III (Version 1.4) is a widely-used, publicly available critical-care dataset from intensive care units at Beth Israel Deaconess Medical Center between 2001 and 2012 [39]. MIMIC-IV (Version 3.1) is a newer release that extends MIMIC-III with updated hospital data from the same institution (2008–2022) [38]. To further reduce potential contamination from public corpora and evaluate language models in a setting less prone to data leakage, we additionally include PROMOTE, a private dataset from Emory Healthcare spanning 2012–2021 [100, 105].

Across all sources, we treat an inpatient stay as the basic encounter unit. For each encounter, we extract billing-code-derived clinical events, including diagnoses and treatments, where treatments cover both medical procedures and medication prescriptions. During preprocessing, we normalize heterogeneous source schemas into a unified event representation with (i) a standardized event type in "{*diagnosis*, *procedure*, *prescription*}", (ii) a human-readable event description mapped from raw clinical codes (e.g., ICD), and (iii) a consistent encounter timeline ordering. We mainly use *PyHealth* [113] to extract information from raw EHRs and perform preprocessing.

From the preprocessed EHRs, we define an *EHR instance* as the minimal input unit presented to the LLM. For the diagnosis and treatment tasks, each instance $u^{(i)}$ corresponds to a single encounter $\mathcal{E}^{(n)}$. For the prognosis task, each instance $u^{(i)}$ corresponds to a pair of consecutive encounters $(\mathcal{E}^{(n)}, \mathcal{E}^{(n+1)})$ from the same patient. Detailed cohort statistics are summarized in Appendix B.

3.3 Template Generation

After preprocessing the EHR cohort, structured templates $\mathcal{P}$ are constructed from $\mathcal{E}$. Each template P_k specifies a clinically grounded blueprint that can be deterministically instantiated into QA items, following the formulation in Section 3.1. Specifically, P_k includes a template context C_k, a target clinical relation $R_k = (x_k, r_k, y_k)$, and latent attributes A_k that support downstream generation. Template construction is implemented through a multi-stage interaction among EHR data, a medical LLM, and a biomedical knowledge base.

Stage 1 (Relation extraction: EHR → LLM → KB). For each EHR input instance $u^{(i)}$, an instruction-fine-tuned medical LLM (specifically, HuatuoGPT-o1-8B [13]) is prompted to extract clinically salient relations from the patient record under strict JSON output constraints. The objective is to capture implicit clinical logic encoded in structured EHR data. Extracted relations are deduplicated and assigned unique identifiers, producing candidate relation triplets $R_k = (x_k, r_k, y_k)$ that later QA items target, such as *(Hyperglycemia, Treat-with, Insulin)*, together with an associated rationale. The LLM is also prompted to extract a small set of auxiliary context events, such as *Tracheostomy* and *Hemothorax*, which are aggregated into C_k. These auxiliary events are constrained to have no lexical or semantic overlap with any entity in R_k, meaning they cannot directly repeat or paraphrase entities appearing in the target relation triplet. Outputs of this stage are passed to KB for verification. More details about the extraction of clinical relations and context from raw EHR data are presented in Appendix C.1.

Stage 2 (Relation verification and enrichment: KB → LLM). In this stage, relations extracted in Stage 1 are validated and enriched through external biomedical evidence by querying a composite KB that integrates UMLS [10], SemMedDB [42], DrugBank [96], PubMed [12], and ICD [67]. UMLS provides standardized concept identifiers (CUIs) and textual definitions across biomedical vocabularies [10]. SemMedDB provides semantic relations (e.g., *Cause* and *Treat-with*) extracted from PubMed abstracts, thereby offering literature-supported evidence for relations between biomedical concepts [42]. Event strings extracted from EHRs are first resolved to standardized concepts through the UMLS API, mapping each entity to a UMLS CUI with source vocabularies such as ICD and DrugBank. After concept linking, evidence for relations is retrieved through SemMedDB.

Given an LLM-extracted clinical relation from a patient EHR record $R_k = (x_k, r_k, y_k)$, we verify its validity by checking for supporting evidence in SemMedDB, which contains KB relation triplets automatically extracted from PubMed abstracts. A candidate relation R_k is retained only if it satisfies all three criteria:

- Positive support: SemMedDB contains evidence supporting the relation, such as $(x_k, \textit{Cause}, y_k)$ or $(x_k, \textit{Treat-with}, y_k)$.
- No negative evidence: SemMedDB does not contain contradictory relations, such as $(x_k, \textit{Neg-cause}, y_k)$.
- No conflicting background evidence: No contradictory relation is found with respect to a predefined set of background concepts C_k, such as $(C_k, \textit{Neg-cause}, y_k)$.

These checks ensure that each retained relation reflects a clinically valid association supported by biomedical knowledge, thereby reducing the risk of hallucinated relations introduced by LLMs.

To support downstream QA generation, entity definitions are retrieved from UMLS, and evidence sentences are retrieved from PubMed through SemMedDB. The verified relations, together with their associated definitions and evidence, are stored in structured templates P_k and used in subsequent QA construction. More details about how to use the KB are provided in Appendix C.2.

Stage 3 (Template completion: LLM → KB). In this stage, each verified template P_k is completed by prompting the LLM to generate additional structured attributes under a strict JSON schema. For each verified relation $R_k = (x_k, r_k, y_k)$, the LLM produces: (i) a set of distractor candidates that compete with the object entity y_k in the relation, e.g., *Magnesium sulfate* and *Furosemide* as distractors for *Insulin*; these candidates are generated by the LLM or sampled from EHR data of other patients to capture both model knowledge and real-world patterns; (ii) a high-level clinical condition topic associated with the relation, e.g., *Hyperglycemia*; and (iii) a concise rationale that summarizes the relation together with KB-retrieved evidence, e.g., *"The prescription of Insulin is likely due to the need to treat Hyperglycemia, as Insulin effectively regulates blood sugar levels."* The resulting attributes are stored in the updated template P_k and used for downstream QA generation. Additional details of the generated templates are provided in Appendix C.3.

Stage 4 (Template filtering: KB → Template Output). In the final stage, unqualified or misleading distractors within templates are removed via KB verification to preserve an unambiguous answer set. Following the procedure from Stage 2, each distractor term is resolved to a UMLS CUI, and SemMedDB is queried for supporting predicate evidence. A distractor is filtered out if it forms any clinically supported relation that would render it a plausible correct answer given the question context. Specifically, a distractor is removed if:

- SemMedDB provides positive evidence linking the distractor to the subject entity x_k under a compatible predicate, such as $(x_k, \text{Cause}, \text{distractor})$.
- SemMedDB provides positive evidence linking the distractor to any auxiliary context event in C_k under a compatible predicate, such as $(C_k, \text{Cause}, \text{distractor})$.

After filtering, between three and five distractors are retained per template; templates failing to meet the minimum distractor count are discarded to ensure QA quality.

3.4 QA Generation

Each template P_k provides (i) a context C_k, (ii) a verified clinical relation $R_k = (x_k, r_k, y_k)$, and (iii) latent attributes A_k, including entity definitions, supporting evidence or rationale, candidate distractors, and an associated clinical topic. The templates are provided to an LLM to instantiate multiple types of QA items.

For each QA item I_j, an event-complete scenario S_j is constructed by augmenting the template context with the relation subject entity, i.e., $S_j \leftarrow C_k \cup \{x_k\}$. This design grounds S_j in observed encounter-level clinical events while explicitly tying the scenario to the verified relation, thereby providing faithful background information for question construction.

Within each template, multiple-choice questions (MCQs) are instantiated with an option count $c \in \{4, 5, 6\}$. For each task, a task-specific question skeleton is used to construct Q_j; for example,

in the prognosis task, the question is phrased as *"Given the prior clinical history summarized above, what diagnosis may occur at the next encounter?"* During evaluation, the tested LLM receives (S_j, Q_j) as input. For a given c, one correct answer is designated and the remaining $c-1$ options are filled with distractors retrieved from the template. The explanation is taken from the template rationale, generated by integrating real-world EHR patterns, internal LLM knowledge, and KB-retrieved evidence. To increase diversity, each question is paraphrased and each choice set is permuted to create multiple MCQ versions. Additional details of the question skeleton and question paraphrasing are provided in Appendix C.4.

Open-ended questions (OEQs) are also constructed to elicit a free-text response and a corresponding explanation aligned with the target clinical relation $R_k = (x_k, r_k, y_k)$. These questions follow the same three clinical decision tasks and task-specific skeletons to form Q_j; for example, an OEQ is phrased as *"Given the prior clinical history summarized above, what event may lead to acute kidney failure at the next encounter? Why?"* The gold-standard answer is defined as the rationale generated during template completion by integrating real-world EHR patterns, internal model knowledge, and KB-retrieved evidence.

For each template P_k and option count c, MCQs are instantiated for 4-choice MCQ (4 paraphrased versions), 5-choice MCQ (5 versions), 6-choice MCQ (6 versions), and 1 OEQ. As a result, each template yields at most 16 QA items across four question types, enabling evaluation under different response constraints. Finally, the generated benchmark contains 960,067 QA items. QA statistics are provided in Appendix B.

4 Experiments

4.1 Benchmarking LLMs on EHRBench Across Clinical Decision Tasks

In our main experiments, we evaluate a comprehensive set of 31 representative LLMs on the constructed benchmark dataset. Details of all utilized LLMs in this study are provided in Appendix J. Specifically, the LLM used for EHRBench generation (HuatuoGPT-o1-8B) is not evaluated here to avoid bias. These evaluated models are categorized into three primary groups:

(a) **Open source general LLMs** that serve as critical performance baselines and widely accessible tools for comparative analysis: (a.1) glm4-9b and glm4-32b [24] (a.2) llama3-8b, llama3-70b, llama3.1-8b, llama3.2-3b, llama3.3-70b [26]; (a.3) mistral-7b, mistral-small3-24b [36], and ministral-8b [53]; (a.4) qwen2.5-3b, qwen2.5-7b and qwen2.5-32b [112]; (a.5) qwen3-4b, qwen3-8b, and qwen3-32b [111]; (a.6) smollm3-3b [7]; (a.7) yi-1.5-9b and yi-1.5-34b [115].

(b) **Medical LLMs** that are pretrained on healthcare corpora and specialized for tackling medical tasks: (b.1) doctor-r1-8b [48]; (b.2) med42-8b [15]; (b.3) ultramedical-8b [117]; (b.4) m1-7b-23k and m1-32b-1k [33].

(c) **HIPAA compliant API-based LLMs** that ensure the secure processing of protected health information: (c.1) gpt-4.1-nano, gpt-4.1-mini, and gpt-4.1 [1]; (c.2) gpt-5-nano, gpt-5-mini, gpt-5, gpt-5.2 [81].

We benchmark 31 representative LLMs on EHRBench to assess their clinical decision-making capability under a unified inference protocol. Specifically, we evaluate three core tasks (Diagnosis/Treatment/Prognosis), three data sources (MIMIC-III/MIMIC-IV/PROMOTE), and three multiple-choice settings (4-choice/5-choice/6-choice). We report accuracy at multiple granularities (task-level, source-level, and type-level), and additionally compute each model's overall accuracy and its rank statistics (mean and standard deviation) across settings, where per-setting ranks are obtained by sorting models by accuracy within each setting and then aggregating ranks across all settings. Further details regarding the experiment protocol, including batching, deterministic decoding, hardware, and the fixed evaluation subset, are provided in Appendix E.1. The aggregated accuracy results are summarized in Table 1. Additional results for cost analysis and error analysis are provided in Appendix E.2 and E.3.

Overall, the benchmark results are broadly consistent with established model capability trends, with the highest-ranked systems corresponding to the most capable and recently released models in our evaluation set, validating the construction pipeline of EHRBench. Specifically, gpt-5.2, gpt-4.1, gpt-5, llama3.3-70b, gpt-4.1-mini, qwen3-32b, glm4-32b, and gpt-5-mini emerge as the strongest performers. Among them, gpt-5.2 achieves the highest overall accuracy of 70.91% with the best average rank of 1.69 and a low rank standard deviation of 1.10, indicating stable performance across tasks, sources, and question types. Meanwhile, the leading open-source models remain highly competitive: llama3.3-70b attains 67.28% and qwen3-32b attains 66.78%, narrowing the gap to API-based models to only 3-4 absolute points. Beyond the leaderboard, the relative ordering within model families follows expected scaling and generation trends. For example, within the Qwen series, qwen3-32b substantially outperforms smaller counterparts such as qwen3-8b (60.87%) and qwen3-4b (60.63%), and also improves over the previous-generation qwen2.5-32b (64.97%). Collectively, these patterns suggest that EHRBench reliably captures meaningful capability differences consistent with model capacity.

Performance varies substantially across clinical decision tasks: treatment selection consistently yields the highest accuracy, whereas prognosis prediction is the most challenging. Overall, the average accuracy across all models and all questions follows the ordering as Tx > Dx > Px (69.33% > 55.02% > 46.67%). This pattern is clinically intuitive. Treatment selection often depends on relatively direct, well-documented associations between medications and their indications, which are explicitly described in drug labels and consolidated in clinical practice guidelines. In contrast, diagnosis and prognosis tasks emphasize disease-to-disease causal and progression relations, which are typically less explicit, more confounded by comorbidities, and harder to infer from limited encounter evidence. Prognosis further requires anticipating conditions beyond the current visit by integrating longitudinal trajectories and subtle risk factors, making it inherently more difficult than in-encounter decision-making. Despite this difficulty, diagnosis completion and especially prognosis are critical for real-world care, underscoring the need to strengthen LLMs for longitudinal reasoning and forward-looking clinical prediction to support clinicians.

Dataset-source-level accuracies exhibit only moderate variation compared with the stronger task- and type-level effects. When

Table 1: Benchmarking LLMs on EHRBench across tasks, data sources, and question types. We use abbreviations Dx/Tx/Px for diagnosis/treatment/prognosis decision task, MIII/MIV/PRO for MIMIC-III/MIMIC-IV/PROMOTE, and 4C/5C/6C for 4/5/6-choice MCQs. Within each column, we mark the top-8 results (ranked first 25%) using underlines and rank superscripts: #1 #2 #3 #4 #5 #6 #7 #8.

Model	Overall	Rank		Task Acc			Source Acc			Type Acc		
	Acc (%)↑	Avg ↓	SD.	Dx (%)	Tx (%)	Px (%)	MIII (%)	MIV (%)	PRO (%)	4C (%)	5C (%)	6C (%)
Open source general LLMs												
glm4-9b	59.62	16.70	2.52	58.36	72.61	47.90	61.92	61.11	56.41	64.76	59.47	54.64
glm4-32b	66.12#7	7.05#7	2.44	67.09#6	77.90#6	53.36	66.27	66.45#8	65.85#4	70.81#8	65.89#8	61.66#7
llama3-8b	48.90	24.23	1.13	44.61	63.44	38.63	51.19	48.60	47.74	54.87	48.00	43.82
llama3-70b	63.35	10.72	3.38	62.21	77.63#7	50.20	65.39	63.72	61.20	68.41	63.19	58.45
llama3.1-8b	56.76	19.41	3.74	53.82	73.45	43.02	57.50	56.77	55.79	62.67	56.66	50.97
llama3.2-3b	49.85	23.67	1.22	43.18	65.41	40.96	50.16	51.20	48.46	55.79	48.78	44.99
llama3.3-70b	67.28#4	5.23#4	2.08	68.35#3	79.05#4	54.44#7	68.74#5	67.94#4	65.35#6	71.98#4	67.07#4	62.79#4
mistral-7b	38.23	28.04	2.13	36.59	41.90	36.21	38.54	38.25	37.85	40.16	38.56	35.98
ministral-8b	56.48	20.32	1.64	53.11	71.97	44.37	58.54	57.39	54.31	61.84	55.79	51.82
mistral-small3-24b	65.01	9.52	4.49	66.20	75.02	53.81#8	67.00#8	65.36	63.38	69.41	64.19	61.42
qwen2.5-3b	37.87	28.98	1.07	34.94	47.75	30.93	39.72	39.21	35.23	47.93	36.09	29.59
qwen2.5-7b	57.74	18.99	2.30	56.27	72.04	44.91	59.76	59.25	54.98	62.22	57.47	53.52
qwen2.5-32b	64.97	8.81	2.95	66.48#7	76.87	51.54	65.22	67.00#7	63.05	69.80	64.50	60.59
qwen3-4b	60.63	14.99	2.77	59.67	73.46	48.76	62.01	61.73	58.39	66.37	60.24	55.27
qwen3-8b	60.87	14.27	2.84	58.19	74.49	49.93	62.26	62.11	58.74	67.09	60.21	55.31
qwen3-32b	66.78#6	6.54#6	2.14	67.97#5	77.34#8	55.04#6	68.48#7	67.18#6	65.55#5	71.33#6	66.47#6	62.55#5
smollm3-3b	45.82	25.84	1.34	40.79	58.29	38.39	46.31	46.08	45.04	51.01	45.09	41.37
yi-1.5-9b	45.51	25.88	1.37	41.97	57.52	37.05	46.87	45.50	44.84	53.15	45.64	37.74
yi-1.5-34b	58.94	17.65	2.65	56.70	72.25	47.86	60.72	60.07	56.56	64.48	58.61	53.71
Medical LLMs												
doctor-r1-8b	61.07	14.06	2.32	58.74	74.49	49.98	61.94	62.26	59.01	67.07	60.56	55.57
med42-8b	36.48	29.31	1.04	33.48	45.54	30.41	36.88	35.00	37.45	38.69	38.41	32.34
ultramedical-8b	29.02	30.60	0.69	19.14	43.09	24.83	31.17	29.56	27.99	38.76	28.26	20.05
m1-7b-23k	46.08	26.01	1.82	38.42	63.07	36.74	46.50	46.99	45.14	50.03	45.75	42.45
m1-32b-1k	63.21	11.47	3.64	63.07	74.84	51.73	62.87	65.49	61.46	68.49	62.80	58.35
HIPAA compliant API-based LLMs												
gpt-4.1-nano	60.48	15.09	2.28	58.02	74.03	49.39	61.59	61.89	58.28	65.42	60.39	55.63
gpt-4.1-mini	66.79#5	6.28#5	1.91	66.41#8	77.90#5	56.05#5	68.58#6	67.36#5	64.45#7	71.34#5	66.76#5	62.26#6
gpt-4.1	69.43#2	2.51#2	1.32	69.87#2	80.10#3	58.33#3	70.59#2	69.77#2	67.87#3	73.97#2	69.21#2	65.11#2
gpt-5-nano	57.80	19.31	1.93	56.39	70.86	46.16	58.64	58.72	55.68	63.27	57.86	52.29
gpt-5-mini	66.12#8	7.84#8	3.48	65.40	76.17	56.79#4	69.08#4	66.05	63.88#8	70.87#7	65.98#7	61.51#8
gpt-5	69.06#3	3.21#3	1.86	68.26#4	80.45#1	58.46#2	70.16#3	69.18#3	68.26#2	73.64#3	68.92#3	64.61#3
gpt-5.2	70.91#1	1.69#1	1.10	72.02#1	80.13#2	60.59#1	71.50#1	71.06#1	70.70#1	75.40#1	70.53#1	66.81#1

aggregating across all tasks and all evaluated models, the accuracies on MIMIC-III/MIMIC-IV/PROMOTE are 58.26%/57.69%/55.45%, suggesting that performance on the two public datasets and the private dataset is broadly comparable. The consistent trends across both public and private data support the robustness of our pipeline, indicating that EHRBench can be instantiated on heterogeneous EHR sources while yielding benchmarks, capturing CDM behaviors with similar difficulty and discriminative power.

Increasing the number of multiple-choice options leads to a clear and consistent decline in accuracy. The average accuracy across all models and all questions decreases from 62.29% (4C) to 56.69% (5C) to 52.04% (6C). Such monotonic trends are expected under a well-constructed multiple-choice benchmark, as additional options increase confusability and reduce the probability of correct selection under uncertainty. The consistent degradation across models therefore provides evidence that the EHRBench pipeline produces valid question instances whose difficulty is appropriately controlled by the number of answer choices.

When comparing medical LLMs to the general-purpose base models they are adapted from, we do not observe consistent gains from medical fine-tuning on EHRBench. For example, m1-32b-1k achieves 63.21% overall accuracy, close to its base model qwen2.5-32b at 64.97%. Appendix E.4 provides a more detailed breakdown. Similar observations have also been reported in prior work [20, 109]. These results suggest that current medical-domain specialization still leaves important gaps for EHR-grounded clinical decision-making. In EHRBench, strong performance requires reasoning

over patients' real longitudinal EHR context and answering questions that demand both biomedical knowledge and nontrivial inference, including disentangling confounded relations such as disease–disease progression and disease–treatment associations. Improving these capabilities likely requires training signals beyond domain text exposure, such as large-scale clinical case supervision and decision-focused objectives, for which EHR-grounded resources like EHRBench may provide a useful foundation.

We also conduct additional analyses to examine whether the main experiment results are sensitive to benchmark construction choices or can be explained by shallow matching heuristics. Specifically, Appendix E.6 evaluates whether the benchmark results are affected by the LLM used for QA generation, and Appendix E.7 studies whether changing the number of local EHR context events alters the observed performance trends. In addition, Appendix E.5 compares LLMs with embedding-based non-LLM retrieval baselines under the same zero-shot QA setting. These analyses show that the main findings are robust to key construction choices and that strong performance on EHRBench cannot be reduced to simple question-option semantic similarity matching.

4.2 Additional Analyses and Validation

Beyond the main benchmark results in Section 4.1, which focus on comparing representative LLMs under a unified zero-shot multiple-choice setting, we conduct several experiments with different settings in the Appendix to validate the reliability of the evaluation protocol and the stability of the main conclusions.

- In Appendix F, we separately benchmark reasoning-oriented LLM configurations because explicit intermediate reasoning substantially increases token usage and would otherwise confound the direct model comparison in Section 4.1. This controlled study characterizes the accuracy-efficiency trade-off under different reasoning-effort settings, showing that additional reasoning generally improves performance while incurring higher token cost. This trend is consistent with expected scaling behavior and further supports the validity of the EHRBench pipeline.
- In Appendix G, we evaluate multiple paraphrased question versions to test whether model performance is sensitive to surface-level wording. The consistently low variability and high prediction consistency across versions indicate that single-version evaluation in the main experiment provides a stable proxy for underlying CDM capability.
- In Appendix H, we run an additional evaluation on the extended question set covering all verified QA templates to examine whether the fixed-subset protocol introduces sampling artifacts. The near-identical model ranking and trend patterns suggest that the selected subset in the main experiment provides sufficient coverage for fair comparison at scale, further supporting the reliability of the generation pipeline.
- In Appendix I, we evaluate paraphrased open-ended questions to probe free-form clinical reasoning beyond multiple-choice selection. The performance trends remain consistent with the main benchmark, supporting the reliability of the open-ended question pipeline.

Collectively, these analyses show that the conclusions drawn from the main benchmark are robust to key evaluation design choices. They also provide additional evidence that EHRBench offers a stable and reliable framework for evaluating EHR-grounded clinical decision-making capabilities.

5 Conclusion

In this work, we develop EHRBench via an automated and reliable pipeline based on *EHR–LLM–KB* interaction. The pipeline (i) converts encounter-level EHR trajectories into structured templates, (ii) deterministically instantiates these templates into large-scale QA items with controlled variants for robust evaluation, and (iii) applies KB-based verification and enrichment to improve reliability.

Under a unified inference protocol, the benchmarking of more than 30 representative LLMs on EHRBench yields consistent trends that validate the benchmark. Recently released high-capability models achieve the strongest performance, treatment selection is consistently easier than the other two tasks, dataset effects remain modest, and current medical fine-tuning does not deliver consistent gains over the corresponding general-purpose base models. Additional analyses further confirm the reliability of the EHRBench construction pipeline and the evaluation protocol used in the main experiments. Collectively, these results provide actionable insights that can inform the design and evaluation of clinically reliable LLM systems in EHR-grounded medical decision making.

Overall, EHRBench provides an automated and reliable benchmark with 960,067 QA items for evaluating LLM-based clinical decision making grounded in real-world structured EHR trajectories. We hope that EHRBench will serve as a practical testbed to accelerate the development of clinically reliable LLM systems and to facilitate transparent and reproducible progress in EHR-grounded medical decision making.

Ethical Statement

This study was conducted in full compliance with established ethical and data governance standards. MIMIC-III and MIMIC-IV are publicly available credentialed datasets accessed under the PhysioNet Credentialed Data Use Agreement and all relevant data usage policies. PROMOTE is a private dataset that was fully de-identified prior to use, and its use was approved by the Emory Institutional Review Board (IRB Protocol 2025P010425).

All raw EHR data were processed locally on HIPAA-compliant systems into structured templates and QA pairs and were not directly exposed to the evaluated LLMs. The released benchmark contains only de-identified QA items rather than raw patient records. We additionally designed the benchmark construction pipeline to reduce information leakage through controlled context generation, KB verification, and filtering procedures. No patient re-identification was attempted at any stage of the study.

Acknowledgement

This research was partially supported by internal funds and GPU resources provided by Emory University, the U.S. National Science Foundation (Awards 2442172, 2312502, and 2319449), and the U.S. National Institutes of Health (Awards K25DK135913, RF1NS139325, R01DK143456, U18DP006922, and R01HL166233).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. 2025. Longhealth: A question answering benchmark with long clinical documents. *Journal of Healthcare Informatics Research* (2025), 1–17.

[3] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925* (2025).

[4] Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, et al. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775* (2025).

[5] Yaara Artsi, Vera Sorin, Eli Konen, Benjamin S Glicksberg, Girish Nadkarni, and Eyal Klang. 2024. Large language models for generating medical examinations: systematic review. *BMC medical education* 24, 1 (2024), 354.

[6] Seongsu Bae, Daeun Kyung, Jaehee Ryu, Eunbyeol Cho, Gyubok Lee, Sunjun Kweon, et al. 2023. Ehrxqa: A multi-modal question answering dataset for electronic health records with chest x-ray images. *Advances in Neural Information Processing Systems* 36 (2023), 3867–3880.

[7] Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patino, et al. 2025. SmolLM3: smol, multilingual, long-context reasoner. *Hugging Face Blog* (2025).

[8] Jayetri Bardhan, Anthony Colas, Kirk Roberts, and Daisy Zhe Wang. 2022. Drugehrqa: A question answering dataset on structured and unstructured electronic health records for medicine related queries. *arXiv preprint arXiv:2205.01290* (2022).

[9] Balu Bhasuran, Qiao Jin, Yuzhang Xie, Carl Yang, Karim Hanna, Jennifer Costa, Cindy Shavor, Wenshan Han, Zhiyong Lu, and Zhe He. 2025. Preliminary analysis of the impact of lab results on large language model generated differential diagnoses. *npj Digital Medicine* 8, 1 (2025), 166.

[10] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.

[11] Joeran S Bosma, Koen Dercksen, Luc Builtjes, Romain André, Christian Roest, Stefan J Fransen, Constant R Noordman, Mar Navarro-Padilla, Judith Lefkes, Natália Alves, et al. 2025. The DRAGON benchmark for clinical NLP. *npj Digital Medicine* 8, 1 (2025), 289.

[12] Kathi Canese and Sarah Weis. 2013. PubMed: the bibliographic database. *The NCBI handbook* 2, 1 (2013), 2013.

[13] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925* (2024).

[14] Christopher Chiu, Silviu Pitis, and Mihaela van der Schaar. 2025. Simulating Viva Voce Examinations to Evaluate Clinical Reasoning in Large Language Models. *arXiv preprint arXiv:2510.10278* (2025).

[15] Clément Christophe, Praveen K Kanithi, Tathagata Raha, Shadab Khan, and Marco AF Pimentel. 2024. Med42-v2: A Suite of Clinical LLMs. https://arxiv.org/abs/2408.06142

[16] Hejie Cui, Alyssa Unell, Bowen Chen, Jason Alan Fries, Emily Alsentzer, Sanmi Koyejo, and Nigam H Shah. 2025. Timer: Temporal instruction modeling and evaluation for longitudinal clinical records. *npj Digital Medicine* 8, 1 (2025), 577.

[17] Amin Dada, Osman Koraş, Marie Bauer, Amanda Butler, Kaleb Smith, Jens Kleesiek, and Julian Friedrich. 2025. Medisumqa: Patient-oriented question-answer generation from discharge letters. In *Proceedings of the Second Workshop on Patient-Oriented Language Processing (CL4Health)*. 124–136.

[18] Wei Dai, Peilin Chen, Malinda Lu, Daniel Li, Haowen Wei, Hejie Cui, and Paul Pu Liang. 2025. Climb: Data foundations for large scale multimodal clinical foundation models. *arXiv preprint arXiv:2503.07667* (2025).

[19] Felix J Dorfner, Amin Dada, Felix Busch, Marcus R Makowski, Tianyu Han, Daniel Truhn, et al. 2024. Biomedical large languages models seem not to be superior to generalist models on unseen medical data. *arXiv preprint arXiv:2408.13833* (2024).

[20] Felix J Dorfner, Amin Dada, Felix Busch, Marcus R Makowski, Tianyu Han, Daniel Truhn, et al. 2025. Evaluating the effectiveness of biomedical fine-tuning for large language models on clinical tasks. *Journal of the American Medical Informatics Association* 32, 6 (2025), 1015–1024.

[21] Yongqi Fan, Hongli Sun, Kui Xue, Xiaofan Zhang, Shaoting Zhang, and Tong Ruan. 2025. Medodyssey: A medical domain benchmark for long context evaluation up to 200k tokens. In *Findings of the Association for Computational Linguistics: NAACL 2025*. 32–56.

[22] Zhihao Fan, Lai Wei, Jialong Tang, Wei Chen, Wang Siyuan, Zhongyu Wei, and Fei Huang. 2025. Ai hospital: Benchmarking large language models in a multi-agent medical interaction simulator. In *Proceedings of the 31st International Conference on Computational Linguistics*. 10183–10213.

[23] Scott L Fleming, Alejandro Lozano, William J Haberkorn, Jenelle A Jindal, Eduardo Reis, Rahul Thapa, Louis Blankemeier, Julian Z Genkins, Ethan Steinberg, Ashwin Nayak, et al. 2024. Medalign: A clinician-generated dataset for instruction following with electronic medical records. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22021–22030.

[24] Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793* (2024).

[25] Linlu Gong, Ante Wang, Yunghwei Lai, Weizhi Ma, and Yang Liu. 2025. The Dialogue That Heals: A Comprehensive Evaluation of Doctor Agents' Inquiry Capability. *arXiv preprint arXiv:2509.24958* (2025).

[26] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[27] Chenlu Guo, Nuo Xu, Yi Chang, and Yuan Wu. 2024. Chbench: A chinese dataset for evaluating health in large language models. *arXiv preprint arXiv:2409.15766* (2024).

[28] Keqi Han, Songlin Zhao, Yao Su, Xiang Li, Yixuan Yuan, Lifang He, and Carl Yang. 2026. Towards a Virtual Neuroscientist: Autonomous Neuroimaging Analysis via Multi-Agent Collaboration. *arXiv preprint arXiv:2605.09366* (2026).

[29] Tessa Han, Aounon Kumar, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Medsafetybench: Evaluating and improving the medical safety of large language models. *Advances in Neural Information Processing Systems* 37 (2024), 33423–33454.

[30] Vinyas Harish, Felipe Morgado, Ariel D Stern, and Sunit Das. 2021. Artificial intelligence and clinical decision making: the new nature of medical uncertainty. *Academic Medicine* 96, 1 (2021), 31–36.

[31] Yutao Hu, Tianbin Li, Quanfeng Lu, Wenqi Shao, Junjun He, Yu Qiao, and Ping Luo. 2024. Omnimedvqa: A new large-scale comprehensive evaluation benchmark for medical lvlm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22170–22183.

[32] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[33] Xiaoke Huang, Juncheng Wu, Hui Liu, Xianfeng Tang, and Yuyin Zhou. 2025. m1: Unleash the potential of test-time scaling for medical reasoning with large language models. *arXiv preprint arXiv:2504.00869* (2025).

[34] Liesbeth Hunik, Asma Chaabouni, Twan van Laarhoven, Tim C Olde Hartman, Ralph TH Leijenaar, Jochen WL Cals, Annemarie A Uijen, and Henk J Schers. 2025. Diagnostic Prediction Models for Primary Care, Based on AI and Electronic Health Records: Systematic Review. *JMIR Medical Informatics* 13, 1 (2025), e62862.

[35] Mingyi Jia, Junwen Duan, Yan Song, and Jianxin Wang. 2025. medikal: Integrating knowledge graphs as assistants of llms for enhanced clinical diagnosis on emrs. In *Proceedings of the 31st International Conference on Computational Linguistics*. 9278–9298.

[36] AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de Las Casas, et al. 2023. Mistral 7B. arXiv 2023. *arXiv preprint arXiv:2310.06825* (2023).

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

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

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

[40] Shreya Johri, Jaehwan Jeong, Benjamin A Tran, Daniel I Schlessinger, Shannon Wongvibulsin, Zhuo Ran Cai, Roxana Daneshjou, and Pranav Rajpurkar. 2024. CRAFT-MD: A conversational evaluation framework for comprehensive assessment of clinical LLMs. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*.

[41] Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad Safranek, Abid Anwar, Andrew Zhang, et al. 2024. Medcalc-bench: Evaluating large language models for medical calculations. *Advances in Neural Information Processing Systems* 37 (2024), 84730–84745.

[42] Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C Rindflesch. 2012. SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics* 28, 23 (2012), 3158–3160.

[43] Michelle Kang Kim, Carol Rouphael, John McMichael, Nicole Welch, and Srinivasan Dasarathy. 2023. Challenges in and opportunities for electronic health record-based data analysis and interpretation. *Gut and liver* 18, 2 (2023), 201.

[44] Yunsoo Kim, Jinge Wu, Yusuf Abdulle, and Honghan Wu. 2024. MedExQA: Medical question answering benchmark with multiple explanations. *arXiv*

preprint arXiv:2406.06331 (2024).

[45] Rachel Knevel and Katherine P Liao. 2023. From real-world electronic health record data to real-world results using artificial intelligence. *Annals of the Rheumatic Diseases* 82, 3 (2023), 306–311.

[46] Yogesh Kumar, Apeksha Koul, Ruchi Singla, and Muhammad Fazal Ijaz. 2023. Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *Journal of ambient intelligence and humanized computing* 14, 7 (2023), 8459–8486.

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

[48] Yunghwei Lai, Kaiming Liu, Ziyue Wang, Weizhi Ma, and Yang Liu. 2025. Doctor-r1: Mastering clinical inquiry with experiential agentic reinforcement learning. *arXiv preprint arXiv:2510.04284* (2025).

[49] Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. 2022. Ehrsql: A practical text-to-sql benchmark for electronic health records. *Advances in Neural Information Processing Systems* 35 (2022), 15589–15601.

[50] Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, et al. 2024. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957* (2024).

[51] Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan Ilgen, Emma Pierson, Pang Wei W Koh, and Yulia Tsvetkov. 2024. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems* 37 (2024), 28858–28888.

[52] Yusheng Liao, Chaoyi Wu, Junwei Liu, Shuyang Jiang, Pengcheng Qiu, Haowen Wang, Yun Yue, Shuai Zhen, Jian Wang, Qianrui Fan, et al. 2025. EHR-R1: A Reasoning-Enhanced Foundational Language Model for Electronic Health Record Analysis. *arXiv preprint arXiv:2510.25628* (2025).

[53] Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, et al. 2026. Ministral 3. *arXiv preprint arXiv:2601.08584* (2026).

[54] Fenglin Liu, Zheng Li, Hongjian Zhou, Qingyu Yin, Jingfeng Yang, Xianfeng Tang, Chen Luo, Ming Zeng, Haoming Jiang, Yifan Gao, et al. 2024. Large language models in the clinic: a comprehensive benchmark. *arXiv preprint arXiv:2405.00716* (2024).

[55] Fenglin Liu, Jinge Wu, Hongjian Zhou, Xiao Gu, Soheila Molaei, Anshul Thakur, Lei Clifton, Honghan Wu, and David A Clifton. 2025. RiskAgent: Autonomous Medical AI Copilot for Generalist Risk Prediction. *arXiv preprint arXiv:2503.03802* (2025).

[56] Jie Liu, Wenxuan Wang, Zizhan Ma, Guolin Huang, Yihang SU, Kao-Jung Chang, Wenting Chen, Haoliang Li, Linlin Shen, and Michael Lyu. 2024. Medchain: Bridging the gap between llm agents and clinical practice through interactive sequential benchmarking. *arXiv preprint arXiv:2412.01605* (2024).

[57] Jie Liu, Wenxuan Wang, Su Yihang, Jingyuan Huang, Yudi Zhang, Cheng-Yi Li, Wenting Chen, Xiaohan Xing, Kao-Jung Chang, Linlin Shen, et al. 2025. Asclepius: A spectrum evaluation benchmark for medical multi-modal large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 24181–24201.

[58] Ruoyu Liu, Kui Xue, Xiaofan Zhang, and Shaoting Zhang. 2025. Interactive evaluation for medical llms via task-oriented dialogue system. In *Proceedings of the 31st International Conference on Computational Linguistics*. 4871–4896.

[59] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126* (2024).

[60] Meng Lu, Yuzhang Xie, Zhenyu Bi, Shuxiang Cao, and Xuan Wang. 2025. CROSSAGENTIE: Cross-Type and Cross-Task Multi-Agent LLM Collaboration for Zero-Shot Information Extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*. 13953–13977.

[61] Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. Expertqa: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 3025–3045.

[62] Izet Masic. 2022. Medical decision making-an overview. *Acta Informatica Medica* 30, 3 (2022), 230.

[63] Nikita Mehandru, Niloufar Golchini, David Bamman, Travis Zack, Melanie F Molina, and Ahmed Alaa. 2025. ER-REASON: A Benchmark Dataset for LLM-Based Clinical Reasoning in the Emergency Room. *arXiv preprint arXiv:2505.22919* (2025).

[64] Benjamin Molinet, Santiago Marro, Elena Cabrio, and Serena Villata. 2024. Explanatory argumentation in natural language for correct and incorrect medical diagnoses. *Journal of Biomedical Semantics* 15, 1 (2024), 8.

[65] Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 10862–10878.

[66] Harsha Nori, Mayank Daswani, Christopher Kelly, Scott Lundberg, Marco Tulio Ribeiro, Marc Wilson, Xiaoxuan Liu, Viknesh Sounderajah, Jonathan Carlson, Matthew P Lungren, et al. 2025. Sequential Diagnosis with Language Models. *arXiv preprint arXiv:2506.22405* (2025).

[67] Kimberly J O'malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. 2005. Measuring diagnoses: ICD code accuracy. *Health services research* 40, 5p2 (2005), 1620–1639.

[68] David Oniani, Xizhi Wu, Shyam Visweswaran, Sumit Kapoor, Shravan Kooragayalu, Katelyn Polanska, and Yanshan Wang. 2024. Enhancing large language models for clinical decision support by incorporating clinical practice guidelines. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*. IEEE, 694–702.

[69] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*. PMLR, 248–260.

[70] Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. *arXiv preprint arXiv:1809.00732* (2018).

[71] Maria Panagioti, Kanza Khan, Richard N Keers, Aseel Abuzour, Denham Phipps, Evangelos Kontopantelis, Peter Bower, Stephen Campbell, Razaan Haneef, Anthony J Avery, et al. 2019. Prevalence, severity, and nature of preventable patient harm across medical care settings: systematic review and meta-analysis. *bmj* 366 (2019).

[72] Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. 2021. Knowledge graph-based question answering with electronic health records. In *Machine Learning for Healthcare Conference*. PMLR, 36–53.

[73] Thierry Pelaccia, Jacques Tardif, Emmanuel Triby, and Bernard Charlin. 2017. A Novel Approach to Study Medical Decision Making in the Clinical Setting: The "Own-point-of-view" Perspective. *Academic Emergency Medicine* 24, 7 (2017), 785–795.

[74] Oriel Perets, Ofir Ben Shoham, Nir Grinberg, and Nadav Rappoport. 2025. CUP-Case: Clinically Uncommon Patient Cases and Diagnoses Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 28293–28301.

[75] Pengcheng Qiu, Chaoyi Wu, Shuyu Liu, Weike Zhao, Zhuoxia Chen, Hongfei Gu, Chuanjin Peng, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2025. Quantifying the reasoning abilities of llms on real-world clinical cases. *arXiv preprint arXiv:2503.04691* (2025).

[76] Preethi Raghavan, Jennifer J Liang, Diwakar Mahajan, Rachita Chandra, and Peter Szolovits. 2021. emrkbqa: A clinical knowledge-base question answering dataset. Association for Computational Linguistics.

[77] Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. 2024. AgentClinic: a multimodal agent benchmark to evaluate AI in simulated clinical environments. *arXiv preprint arXiv:2405.07960* (2024).

[78] Mingchen Shao, Yuzhang Xie, Carl Yang, and Jiaying Lu. 2026. LLM-MINE: Large Language Model based Alzheimer's Disease and Related Dementias Phenotypes Mining from Clinical Notes. *arXiv preprint arXiv:2603.13673* (2026).

[79] Ofir Ben Shoham and Nadav Rappoport. 2024. MedConceptsQA: Open source medical concepts QA benchmark. *Computers in Biology and Medicine* 182 (2024), 109089.

[80] Damien Sileo, Kanimozhi Uma, and Marie Francine Moens. 2024. Generating multiple-choice questions for medical question answering with distractors and cue-masking. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 7647–7653.

[81] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267* (2025).

[82] 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. 2023. Large language models encode clinical knowledge. *Nature* 620, 7972 (2023), 172–180.

[83] Vivek Subbiah. 2023. The next generation of evidence-based medicine. *Nature medicine* 29, 1 (2023), 49–58.

[84] Yu Sun, Xingyu Qian, Weiwen Xu, Hao Zhang, Chenghao Xiao, Long Li, Deli Zhao, Wenbing Huang, Tingyang Xu, Qifeng Bai, et al. 2025. Reasonmed: A 370k multi-agent generated dataset for advancing medical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 26457–26478.

[85] Tao Tu, Mike Schaekermann, Anil Palepu, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Yong Cheng, et al. 2025. Towards conversational diagnostic artificial intelligence. *Nature* (2025), 1–9.

[86] Baptiste Vasey, Stephan Ursprung, Benjamin Beddoe, Elliott H Taylor, Neale Marlow, Nicole Bilbro, Peter Watkinson, and Peter McCulloch. 2021. Association of clinician diagnostic performance with machine learning–based decision

support systems: a systematic review. *JAMA network open* 4, 3 (2021), e211276–e211276.

[87] Ping Wang, Tian Shi, Khushbu Agarwal, Sutanay Choudhury, and Chandan K Reddy. 2022. Attention-based aspect reasoning for knowledge base question answering on clinical notes. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–6.

[88] Ping Wang, Tian Shi, and Chandan K Reddy. 2020. Text-to-SQL generation for question answering on electronic medical records. In *Proceedings of The Web Conference 2020*. 350–361.

[89] Ruiyu Wang, Yuzhang Xie, Xiao Hu, Carl Yang, and Jiaying Lu. 2025. BioMedJImpact: A Comprehensive Dataset and LLM Pipeline for AI Engagement and Scientific Impact Analysis of Biomedical Journals. *arXiv preprint arXiv:2511.12821* (2025).

[90] Xiaoda Wang, Ching Chang, Defu Cao, Kaiqiao Han, Fang Sun, Yue Huang, Minxiao Wang, Chang Xu, Xiao Luo, Runze Yan, et al. 2026. Position: Beyond prediction: Toward verifiable physiological waveform reasoning with foundation models and agentic llms. (2026).

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

[92] Xidong Wang, Nuo Chen, Junyin Chen, Yidong Wang, Guorui Zhen, Chunxian Zhang, Xiangbo Wu, Yan Hu, Anningzhe Gao, Xiang Wan, et al. 2024. Apollo: A lightweight multilingual medical LLM towards democratizing medical AI to 6B people. *arXiv preprint arXiv:2403.03640* (2024).

[93] Xiaoda Wang, Kaiqiao Han, Yuhao Xu, Xiao Luo, Yizhou Sun, Wei Wang, and Carl Yang. 2026. SE-Diff: Simulator and Experience Enhanced Diffusion Model for Comprehensive ECG Generation. In *The Fourteenth International Conference on Learning Representations*.

[94] Zifeng Wang, Qiao Jin, Jiacheng Lin, Junyi Gao, Jathurshan Pradeepkumar, Pengcheng Jiang, Benjamin Danek, Zhiyong Lu, and Jimeng Sun. 2025. TrialPanorama: Database and Benchmark for Systematic Review and Design of Clinical Trials. *arXiv preprint arXiv:2505.16097* (2025).

[95] Mingyang Wei, Dehai Min, Zewen Liu, Yuzhang Xie, Guanchen Wu, Ziyang Zhang, Carl Yang, Max SY Lau, Qi He, Lu Cheng, et al. 2026. EpiQAL: Benchmarking Large Language Models in Epidemiological Question Answering for Enhanced Alignment and Reasoning. *arXiv preprint arXiv:2601.03471* (2026).

[96] David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. 2008. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research* 36, suppl_1 (2008), D901–D906.

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

[98] Chaoyi Wu, Pengcheng Qiu, Jinxin Liu, Hongfei Gu, Na Li, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2025. Towards evaluating and building versatile large language models for medicine. *npj Digital Medicine* 8, 1 (2025), 58.

[99] Guanchen Wu, Yuzhang Xie, Huanwei Wu, Zhe He, Hui Shao, Xiao Hu, and Carl Yang. 2025. Utilizing Large Language Models for Zero-Shot Medical Ontology Extension from Clinical Notes. *arXiv preprint arXiv:2511.16548* (2025).

[100] Yuhua Wu, Fadi Nahab, Yi Ge, Yuzhang Xie, Hassan Aboul-Nour, Carl Yang, and Xiao Hu. 2025. Prediction of post-stroke AF in ESUS patients is enhanced by combining expert-derived predictors and embedding of full diagnostic codes using pre-trained hypergraph neural networks. *STROKE* 56 (2025).

[101] Yunpeng Xiao, Carl Yang, Mark Mai, Xiao Hu, and Kai Shu. 2025. Beyond MedQA: Towards Real-world Clinical Decision Making in the Era of LLMs. *arXiv preprint arXiv:2510.20001* (2025).

[102] Yuzhang Xie, Hejie Cui, Ziyang Zhang, Jiaying Lu, Kai Shu, Fadi Nahab, Xiao Hu, and Carl Yang. 2025. KERAP: A Knowledge-Enhanced Reasoning Approach for Accurate Zero-shot Diagnosis Prediction Using Multi-agent LLMs. *arXiv preprint arXiv:2507.02773* (2025).

[103] Yuzhang Xie, Xu Han, Ran Xu, Xiao Hu, Jiaying Lu, and Carl Yang. 2025. HypKG: Hypergraph-Based Knowledge Graph Contextualization for Precision Healthcare. In *International Semantic Web Conference*. Springer, 328–348.

[104] Yuzhang Xie, Jiaying Lu, Joyce Ho, Fadi Nahab, Xiao Hu, and Carl Yang. 2024. PromptLink: leveraging large language models for cross-source biomedical concept linking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2589–2593.

[105] Yuzhang Xie, Fadi Nahab, Yi Ge, Yuhua Wu, Jessica Saurman, Carl Yang, and Xiao Hu. 2025. Abstract WP175: Predicting Post-Stroke Cognitive Impairment (PSCI) Using Multiple Machine Learning Approaches. *Stroke* 56, Suppl_1 (2025), AWP175–AWP175.

[106] Yuzhang Xie, Guoshuai Niu, Qian Da, Wentao Dai, and Yang Yang. 2022. Survival prediction for gastric cancer via multimodal learning of whole slide images and gene expression. In *2022 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 1311–1316.

[107] Yuzhang Xie, Qingqing Sang, Qian Da, Guoshuai Niu, Shijie Deng, Haoran Feng, Yunqin Chen, Yuan-Yuan Li, Bingya Liu, Yang Yang, et al. 2024. Improving diagnosis and outcome prediction of gastric cancer via multimodal learning using whole slide pathological images and gene expression. *Artificial intelligence in medicine* 152 (2024), 102871.

[108] Yuzhang Xie, Yuhua Wu, Ruiyu Wang, Fadi Nahab, Xiao Hu, and Carl Yang. 2026. Enhanced Atrial Fibrillation Prediction in ESUS Patients with Hypergraph-based Pre-training. *arXiv preprint arXiv:2603.13297* (2026).

[109] Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Xiangru Tang, Hang Wu, May Dongmei Wang, Peifeng Ruan, Donghan Yang, Tao Wang, et al. 2025. Medagentgym: Training llm agents for code-based medical reasoning at scale. In *The Second Workshop on GenAI for Health: Potential, Trust, and Policy Compliance*.

[110] Lawrence KQ Yan, Qian Niu, Ming Li, Yichao Zhang, Caitlyn Heqi Yin, Cheng Fei, Benji Peng, Ziqian Bi, Pohsun Feng, Keyu Chen, et al. 2024. Large language model benchmarks in medical tasks. *arXiv preprint arXiv:2410.21348* (2024).

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

[112] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115* (2024).

[113] Chaoqi Yang, Zhenbang Wu, Patrick Jiang, Zhen Lin, Junyi Gao, Benjamin Danek, and Jimeng Sun. 2023. PyHealth: A Deep Learning Toolkit for Healthcare Predictive Modeling. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) 2023*. https://github.com/sunlabuiuc/PyHealth

[114] Jin Ye, Guoan Wang, Yanjun Li, Zhongying Deng, Wei Li, Tianbin Li, Haodong Duan, Ziyan Huang, Yanzhou Su, Benyou Wang, et al. 2024. Gmai-mmbench: A comprehensive multimodal evaluation benchmark towards general medical ai. *Advances in Neural Information Processing Systems* 37 (2024), 94327–94427.

[115] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652* (2024).

[116] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9556–9567.

[117] Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, et al. 2024. Ultramedical: Building specialized generalists in biomedicine. *Advances in Neural Information Processing Systems* 37 (2024), 26045–26081.

[118] Ming Zhang, Yujiong Shen, Zelin Li, Huayu Sha, Binze Hu, Yuhui Wang, Chenhao Huang, Shichun Liu, Jingqi Tong, Changhao Jiang, et al. 2025. LLMEval-Med: A Real-world Clinical Benchmark for Medical LLMs with Physician Validation. *arXiv preprint arXiv:2506.04078* (2025).

[119] Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415* (2023).

[120] Ziyang Zhang, Hejie Cui, Ran Xu, Yuzhang Xie, Joyce C Ho, and Carl Yang. 2024. Tacco: Task-guided co-clustering of clinical concepts and patient visits for disease subtyping based on ehr data. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6324–6334.

[121] Ziyang Zhang, WANG Lily, MENG Weimin, LIU Chang, SHAO Hui, V SUN Yan, GUO Jingchuan, BIAN Jiang, YIN Rui, and YANG Carl. 2025. Type 2 Diabetes Subtyping via Phenotype and Genotype Co-Learning. *Studies in health technology and informatics* 329 (2025), 1064.

[122] Shuang Zhou, Wenya Xie, Jiaxi Li, Zaifu Zhan, Meijia Song, Han Yang, Cheyenna Espinoza, Lindsay Welton, Xinnie Mai, Yanwei Jin, et al. 2025. Automating expert-level medical reasoning evaluation of large language models. *npj Digital Medicine* (2025).

[123] Yuxuan Zhou, Xien Liu, Chen Ning, and Ji Wu. 2024. Multifaceteval: Multifaceted evaluation to probe llms in mastering medical knowledge. *arXiv preprint arXiv:2406.02919* (2024).

[124] Z. Zhou et al. 2025. Large language models for disease diagnosis: a scoping review. *NPJ Digital Medicine* (2025).

[125] Yakun Zhu, Zhongzhen Huang, Linjie Mu, Yutong Huang, Wei Nie, Jiaji Liu, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. 2025. DiagnosisArena: Benchmarking Diagnostic Reasoning for Large Language Models. *arXiv preprint arXiv:2505.14107* (2025).

[126] Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. 2025. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362* (2025).

Appendices

A Notation Table

Table 2: Notations used in the paper.

Notation	Description
$\mathcal{E}$	Cohort of structured EHR encounter records.
N	Number of encounters in the cohort $\mathcal{E}$.
$\mathcal{E}^{(n)}$	Structured EHR record for encounter n.
$\pi(n)$	Patient identifier associated with encounter n.
$e_m^{(n)}$	The m-th clinical event in encounter n.
M_n	Number of events observed in encounter n.
$d_m^{(n)}$	Textual description of event $e_m^{(n)}$ (e.g., diagnosis/prescription).
$t_m^{(n)}$	Timestamp of event $e_m^{(n)}$.
$a_m^{(n)}$	Additional attributes of event $e_m^{(n)}$ (e.g., codes / numeric values).
g	Template generation function mapping $\mathcal{E}$ to template set $\mathcal{P}$.
$\mathcal{P}$	Set of constructed QA templates.
K	Number of templates in $\mathcal{P} = \{P_k\}_{k=1}^{K}$.
P_k	The k-th QA template.
C_k	Template context set (selected events) for template P_k.
$R_k = (x_k, r_k, y_k)$	Verified clinical relation in template P_k.
x_k	Subject entity of relation R_k.
r_k	Relation predicate of R_k.
y_k	Object entity of relation R_k.
A_k	Latent attributes associated with template P_k (definitions/evidence/distractors/topic).
f	QA instantiation function mapping templates $\mathcal{P}$ to QA items $\mathcal{I}$.
$\mathcal{I}$	Set of constructed QA items.
J	Number of QA items in $\mathcal{I} = \{(S_j, Q_j, B_j)\}_{j=1}^{J}$.
S_j	Scenario (natural-language background) of QA item j.
Q_j	Question text of QA item j.
B_j	Metadata bundle of QA item j (choices/answer/rationale/topic/relations).
$\mathcal{D}^{(n)}$	Set of diagnoses observed in encounter n.
$\mathcal{T}^{(n)}$	Set of treatments (prescriptions/procedures) observed in encounter n.
$\mathcal{S}^{(n)}$	Scenario event subset used to form the QA context for encounter n.
$d_{\text{tgt}}^{(n)}$	Target (withheld) diagnosis in encounter n for diagnosis decision.
$t_{\text{tgt}}^{(n)}$	Target treatment in encounter n for treatment decision.
$d_{\text{tgt}}^{(n+1)}$	Target diagnosis in encounter n+1 for prognosis decision.
$p(\cdot \mid \mathcal{S}^{(n)})$	Conditional distribution of the target event given scenario $\mathcal{S}^{(n)}$.
$\lvert C_k \rvert$	Cardinality of the context set C_k.
CAUSE_REL	Positive SemMedDB predicate group used to verify “cause” relations.
AFFECT_REL	Positive SemMedDB predicate group used to verify “affect” relations.
ASSOC_REL	Positive SemMedDB predicate group used to verify “associate-with” relations.
USAGE_REL	Positive SemMedDB predicate group used to verify treatment-usage relations.
NEG_CAUSE_REL	Negative SemMedDB predicate group used for contradiction checks of “cause” relations.
NEG_AFFECT_REL	Negative SemMedDB predicate group used for contradiction checks of “affect” relations.
NEG_ASSOC_REL	Negative SemMedDB predicate group used for contradiction checks of “associate-with” relations.
NEG_USAGE_REL	Negative SemMedDB predicate group used for contradiction checks of treatment-usage relations.

B EHRBench QA Statistics

For EHRBench, raw EHR records are drawn from three real-world sources: MIMIC-III, MIMIC-IV, and PROMOTE. MIMIC-III (Version 1.4) is a publicly available critical-care dataset containing 38,597 distinct patients and 53,423 hospital admissions from intensive care units at Beth Israel Deaconess Medical Center between 2001 and 2012 [39]. MIMIC-IV (Version 3.1) is a newer release that extends MIMIC-III with updated hospital data from the same institution (2008–2022), including 364,627 patients and 546,028 hospital admissions [38]. To further reduce potential contamination from public corpora and to evaluate language models in a setting less prone to data leakage, PROMOTE is additionally included as a private dataset from Emory Healthcare spanning 2012–2021, with records for 18,561 patients and 912,706 clinical records [100, 105].

To ensure that each constructed QA item is grounded in sufficiently informative clinical documentation, we retain only encounters with rich structured signals. Concretely, an encounter is included only if it contains at least five diagnosis events and at least three treatment events (where treatments include both prescriptions and procedures). This filtering reduces the risk of constructing questions from sparse or under-specified visits and improves the reliability of the downstream relation extraction and QA instantiation.

For each data source and task, up to the first 10,000 EHR instances are extracted and processed by the construction pipeline. End-to-end construction requires approximately 560 H100 GPU-hours. Across all sources and tasks, 465,748 candidate clinical relations are first extracted from EHR trajectories, and 62,786 knowledge-base-verified QA templates are retained (13.5% retention). These templates are then instantiated into 960,067 QA items spanning three decision-making tasks—diagnosis (Dx), treatment (Tx), and prognosis (Px)—and four question formats: 4/5/6-choice multiple-choice questions (MCQs) and open-ended questions (OEQs). Table 3 summarizes the breakdown by data source and task. Treatment accounts for the largest share of questions (450,501), followed by prognosis (259,123) and diagnosis (250,443). Aggregated by data source, EHRBench contains 323,193 questions from MIMIC-III, 259,089 from MIMIC-IV, and 377,785 from PROMOTE, totaling **960,067** questions.

Table 3: EHRBench construction statistics by task, data source, and question format. "Candidate Relations" counts relation candidates extracted from EHR trajectories before knowledge-base (KB) verification, and "Verified Templates" counts KB-supported templates retained for instantiation. "MCQ-4/5/6" report the numbers of 4/5/6-choice MCQs, and "OEQ" reports the number of open-ended questions.

Task	Data Source	Candidate Relations	Verified Templates	MCQ-4	MCQ-5	MCQ-6	OEQ	Total
Diagnosis	MIMIC-III	32,338	4,406	17,624	21,315	24,264	4,329	67,532
	MIMIC-IV	36,869	4,323	17,288	21,085	24,438	4,291	67,102
	PROMOTE	56,699	7,259	29,024	36,250	43,278	7,257	115,809
	Task-Total	125,906	15,988	63,936	78,650	91,980	15,877	250,443
Treatment	MIMIC-III	53,637	9,878	39,428	49,230	58,926	9,873	157,457
	MIMIC-IV	38,140	8,178	32,672	40,775	48,762	8,170	130,379
	PROMOTE	45,899	10,173	40,672	50,830	60,990	10,173	162,665
	Task-Total	137,676	28,229	112,772	140,835	168,678	28,216	450,501
Prognosis	MIMIC-III	71,896	7,207	28,784	30,820	31,404	7,196	98,204
	MIMIC-IV	60,816	4,452	17,800	19,470	19,890	4,448	61,608
	PROMOTE	69,454	6,910	27,632	31,325	33,444	6,910	99,311
	Task-Total	202,166	18,569	74,216	81,615	84,738	18,554	259,123
Full	**EHRBench**	465,748	62,786	250,924	301,100	345,396	62,647	**960,067**

From the length statistics, each question stem is relatively detailed, averaging 332.4 characters (about 46.8 words). The answer options are much more concise: each option averages 28.6 characters (about 3.5 words), indicating that choices are typically short concept-level phrases. The reason field is also substantial, averaging 251.1 characters (about 36.4 words), suggesting there is enough space for coherent justifications rather than single-sentence fragments.

C Details of Methodology

This appendix provides additional implementation details for the benchmark construction pipeline described in Section 3, including encounter filtering, relation/context constraints, concept linking via UMLS, evidence retrieval via SemMedDB, task-specific relation definitions, and distractor generation and verification.

C.1 Clinical Relations and Context Extraction from EHR

Relation cap. For each EHR instance, the prompting stage may yield multiple candidate relations. To control template complexity and limit noise from overly long candidate lists, we retain at most 15 candidate relations per instance after deduplication.

Context construction and size. For each retained relation template P_k, we generate a small auxiliary context set C_k consisting of exactly two events per instance, as described in Section 3.3. During QA instantiation, each question scenario incorporates three events in total: the two context events plus the relation-subject entity, i.e.,

$$S_j \leftarrow C_k \cup \{x_k\}, \quad \text{with } |C_k| = 2. \tag{13}$$

This design yields concise, controlled scenarios while preserving grounding in observed encounter events.

Information leakage prevention. When extracting clinical relations and contextual events from raw EHR data, we enforce strict non-overlap constraints between context events and clinical relations to prevent information leakage, since they may be developed into question stems and choices. This design ensures that each generated question is clinically meaningful rather than surface-level pattern matching.

Task-Specific Relation Definitions. Each template contains a verified relation $R_k = (x_k, r_k, y_k)$ and is associated with one of three clinical decision tasks. We define the permissible entity roles and relation label spaces as follows.

(I) Diagnosis decision. For the diagnosis task, both x_k and y_k are diagnoses recorded in the *same* encounter. LLMs will focus on relations reflecting shared pathophysiologic mechanisms, or common comorbidity patterns, supporting questions that ask which additional diagnosis is likely to be identified given other diagnoses already present in the encounter. The relation label is constrained to the same set:

$$r_k \in \{\textit{cause},\ \textit{affect},\ \textit{associate-with}\}.$$

(II) Treatment decision. For the treatment task, x_k is a diagnosis from the encounter and y_k is a treatment action (prescription or procedure) from the same encounter. The relation label is constrained to:

$$r_k \in \{\textit{treat-with-drug},\ \textit{treat-with-procedure}\},$$

with evidence matched through *USAGE_REL* predicates and entity typing determined by the event category (prescription vs. procedure) after preprocessing.

(III) Prognosis decision. For the prognosis task, x_k is a diagnosis or treatment from the *prior* encounter, and y_k is a diagnosis in the *subsequent* encounter. LLMs will focus on direct complications, risk-modifying effects, or clinically plausible associations that link earlier conditions or interventions to later outcomes. The relation label is constrained to

$$r_k \in \{\textit{cause},\ \textit{affect},\ \textit{associate-with}\},$$

where these labels map to SemMedDB predicate groups via *CAUSE_REL*, *AFFECT_REL*, and *ASSOC_REL* (and their negative counterparts for contradiction checks).

C.2 KB Usage

Concept linking. Concept linking is performed using the UMLS RESTful web services (UMLS web search) provided by the UMLS API.[2] The current UMLS web version at query time is used. Each entity mention extracted from EHRs or produced during template completion (e.g., relation entities and distractor candidates) is resolved to a UMLS Concept Unique Identifier (CUI) using UMLS search.

To improve mapping precision, the search space is constrained to source vocabularies when appropriate:

- For **prescriptions**, source vocabularies aligned with **DrugBank** are prioritized.
- For **procedures** and **diagnoses**, source vocabularies aligned with **ICD** are prioritized.

The resolved CUIs serve as canonical identifiers for downstream KB retrieval and predicate matching in SemMedDB.

SemMedDB Dataset. The SemMedDB dataset is downloaded[3], and the *PREDICATION* and *SENTENCE* files are used for relation verification and enrichment. In the *PREDICATION* file, duplicate records are removed based on the *SUBJECT*, *OBJECT*, and *PREDICATE* columns to obtain unique KB relations. Predicates are then grouped into positive and negative sets that correspond to the target labels for verification

[2]UMLS API Link
[3]SemMedDB Link

and filtering:

$$CAUSE\_REL = \{CAUSES,\ PRODUCES,\ CONVERTS\_TO\ \},$$

$$AFFECT\_REL = \left\{ \begin{matrix} PREDISPOSES,\ COMPLICATES,\ STIMULATES, \\ AUGMENTS,\ AFFECTS \end{matrix} \right\},$$

$$ASSOC\_REL = \{ASSOCIATED\_WITH,\ COEXISTS\_WITH,\ INTERACTS\_WITH\ \},$$

$$USAGE\_REL = \left\{ \begin{matrix} TREATS,\ DIAGNOSES,\ ADMINISTERED\_TO,\ USED\_FOR, \\ MEASUREMENT\_OF,\ METHOD\_OF,\ PREVENTS,\ INHIBITS,\ DISRUPTS \end{matrix} \right\},$$

and their corresponding negative forms:

$$NEG\_CAUSE\_REL = \{NEG\_CAUSES,\ NEG\_PRODUCES,\ NEG\_CONVERTS\_TO\ \},$$

$$NEG\_AFFECT\_REL = \left\{ \begin{matrix} NEG\_PREDISPOSES,\ NEG\_COMPLICATES,\ NEG\_STIMULATES, \\ NEG\_AUGMENTS,\ NEG\_AFFECTS \end{matrix} \right\},$$

$$NEG\_ASSOC\_REL = \{NEG\_ASSOCIATED\_WITH,\ NEG\_COEXISTS\_WITH,\ NEG\_INTERACTS\_WITH\ \},$$

$$NEG\_USAGE\_REL = \left\{ \begin{matrix} NEG\_TREATS,\ NEG\_DIAGNOSES,\ NEG\_ADMINISTERED\_TO,\ NEG\_USED\_FOR, \\ NEG\_MEASUREMENT\_OF,\ NEG\_METHOD\_OF,\ NEG\_PREVENTS,\ NEG\_INHIBITS,\ NEG\_DISRUPTS \end{matrix} \right\}.$$

After evidence is identified, the corresponding *SENTENCE_ID* is used to retrieve supporting PubMed sentences from the *SENTENCE* file.

C.3 QA Template Completion

Distractor generation. For each verified relation template, an LLM generates 10 distractor candidates based on internal knowledge and structured template attributes. In addition, observed clinical events are sampled to expand the candidate pool, yielding up to 25 total distractor candidates per template.

Topic generation. For each relation template, a high-level clinical condition topic is generated to summarize the clinical focus of the relation. For diagnosis and prognosis tasks, the topic is derived from y_k (the object entity). For treatment tasks, the topic is derived from x_k (the subject entity), because the object entity is a treatment rather than a diagnosis. Statistics of the generated topics are reported in Appendix D.

Rationale generation. The LLM is prompted to review the extracted clinical relations and the retrieved KB evidence (e.g., entity definitions, positive evidence, and supporting sentences) to generate a concise clinical rationale. The resulting rationales are stored in the templates and can be directly used as explanations for the MCQs and OEQs derived from the corresponding templates.

Template diversity constraint per patient. To avoid over-representing a single patient trajectory and to promote diversity across conditions and care patterns, the number of verified templates per patient is restricted. Each patient contributes at most three templates across all eligible encounters, selected to maximize coverage of distinct relations and reduce redundancy.

Minimum distractor threshold. To ensure that MCQs include meaningful alternatives and reduce ambiguity, templates with fewer than three verified distractors after filtering are discarded. As a result, each instantiated MCQ contains 4, 5, or 6 choices. This constraint is applied before the final QA instantiation stage.

C.4 QA Generation

Question Skeleton.

- Diagnosis MCQ: Based on the clinical context summarized above, which additional diagnosis is most likely to be present or identified during this visit?
- Treatment MCQ: Given the clinical context summarized above, which treatment is most likely to be prescribed during this visit?
- Prognosis MCQ: Given the prior clinical history summarized above, which diagnosis is most likely to be present or identified at the next visit?
- Diagnosis OEQ: Given the diagnoses of the patient, what may lead to this target diagnosis during the same visit? Why?
- Treatment OEQ: Given the diagnoses of the patient, what may lead to this target treatment during the same visit? Why?

- Prognosis OEQ: Given the prior visit of the patient, what may lead to this target diagnosis at the next visit? Why?

Question Paraphrasing. To improve robustness and linguistic diversity, an instruction-following LLM is used to generate V paraphrased versions of the same scenario while preserving clinical intent. Concretely, V=6 paraphrased scenarios $\{S_j^{(v)}\}_{v=1}^{V}$ are generated, and each scenario is paired with a fixed ask-only question to form the corresponding question instances. Prompts are designed to produce surface-level paraphrases of the same clinical context by rephrasing wording and sentence structure without introducing new information, clinical interpretation, or emphasis across events. Furthermore, the LLM is prohibited from using abstraction or causal language, and similar length and structure are maintained across versions to ensure that variation is limited to linguistic form rather than semantic content. This controlled paraphrasing strategy enables robust evaluation under deterministic rewording while preventing information leakage and preserving clinical meaning.

Choice Permutation. To mitigate sensitivity to answer-position bias, c permutation variants are generated for each instance using a fixed random seed to ensure reproducibility. For a c-choice MCQ, each variant permutes the order of the c answer options while updating the ground-truth label to match the new position of the correct answer. As a result, across the c variants, each option appears exactly once in each position, eliminating systematic advantages associated with any specific answer index. This design prevents models from exploiting positional heuristics, enforces position-invariant evaluation, and enables a more faithful assessment of model reasoning ability rather than sensitivity to answer ordering.

C.5 Trade-off in Knowledge-base Verification

The benchmark construction pipeline adopts a precision-first verification strategy: uncertain or weakly supported relations are discarded rather than retained. This design choice is intended to improve the reliability and clinical validity of retained QA items at scale. Among candidate relations not retained, approximately 81% lacked supporting KB evidence, 12% were associated with explicit negative evidence, 5% were removed by the per-patient diversity cap, and 2% failed distractor-quality filtering. These statistics indicate that most rejected candidates arise from insufficient supporting evidence rather than direct contradiction.

To estimate potential false negatives introduced by the conservative filtering strategy, we manually audited 225 rejected candidate relations (25 sampled items across 3 tasks and 3 data sources). Approximately 24% were judged clinically plausible despite not being retained. These misses likely arise from incomplete KB coverage for medically reasonable relations, near-synonymous CUI granularity mismatch, and entity-linking loss introduced during API-based normalization.

This behavior reflects the intended trade-off of the current pipeline. EHRBench prioritizes higher precision and stronger grounding of retained relations over maximizing recall, acknowledging that some clinically plausible but weakly supported relations may be excluded during verification.

C.6 Quality Control

LLM-based generation enables scalable and automated benchmark construction; however, reliability in the clinical domain requires systematic safeguards. We therefore implement multi-stage quality control throughout the construction of EHRBench to improve correctness, clarity, and fairness. These safeguards are tightly integrated into the *EHR–LLM–KB* pipeline and operate at multiple stages of template generation, instantiation, and validation. The quality control pipeline consists of the following components:

- **Terminology normalization.** Raw EHR events are mapped to normalized, human-readable descriptions and standardized biomedical codes (UMLS CUIs). This step bridges heterogeneity across EHR systems and external knowledge bases, reducing spurious variation caused by synonyms, abbreviations, and data-source-specific naming conventions. Normalization ensures consistent grounding of clinical concepts across data sources and enables reliable downstream KB verification.
- **Knowledge-base verification of clinical relations.** To reduce hallucinations and improve clinical validity, all LLM-extracted relations are verified and enriched using external biomedical knowledge bases, including UMLS, SemMedDB, PubMed, ICD, and DrugBank. Relations are retained only if they are supported by positive evidence and are not contradicted by known negative or conflicting relations. Unsupported or contradictory relations are discarded during Stage 2 of template generation, ensuring that retained relations reflect clinically plausible associations grounded in biomedical knowledge.
- **Knowledge-base filtering of templates.** To preserve an unambiguous choice structure in multiple-choice questions, candidate distractors are further pruned through KB-based filtering. Clinically plausible alternatives that are also supported by knowledge bases given the anchor concept and auxiliary context are removed, reducing the risk of multiple correct answers. This filtering step, applied in Stage 4 of template generation, enforces single-answer correctness while maintaining clinical realism.
- **Leakage prevention.** To minimize information leakage from the context to the answer options, we enforce forbidden-term constraints and non-overlap rules between question stems and choices. In particular, extracted clinical relations are required not to trivially overlap with events explicitly mentioned in the EHR context. These constraints prevent models from exploiting surface cues or lexical overlap instead of performing genuine clinical reasoning.
- **Format and structural validation.** All LLM outputs are required to conform to a predefined JSON schema and are parsed using robust validation routines. At the QA-item level, each record is checked to ensure that required fields are present and non-empty,

that option sets are well-formed, and that duplicate or malformed items are removed. This validation step guarantees structural consistency across the entire benchmark.

- **Diversity and fairness controls.** EHRBench is constructed at large scale, spanning multiple clinical decision tasks, sources, and question types, resulting in a total of 960,067 QA items. To mitigate evaluation bias, multiple deterministic variants are generated through paraphrasing and answer-option permutations, ensuring that model performance reflects robust reasoning rather than sensitivity to surface form. This design promotes fair and stable comparison across models.

Together, these safeguards ensure that the generated QA items fully leverage the scalability of LLM-based generation and the richness of real-world EHR data, while maintaining high standards of clinical correctness, structural clarity, and evaluation fairness.

C.7 EHRBench Usage Protocol

The generated QA items are used to evaluate the CDM capability of LLMs. For each QA item $I_j = (S_j, Q_j, B_j)$, the model receives only the natural-language input (S_j, Q_j). All metadata B_j, including the answer, choices, evidence, and clinical relations, are withheld from the model and used exclusively for scoring and post-hoc analysis. The model is required to produce an answer in the specified format for the corresponding question type.

Because the benchmark contains parallel variants and multiple question types, flexible evaluation settings are supported. For efficiency and broader coverage of clinical concepts, evaluation can be conducted on a fixed subset of variants, such as a single version of the 4-choice questions (e.g., Version 1), which provides a controlled and non-redundant evaluation set. For robustness, evaluation can be conducted across multiple variants and aggregated, for example, by combining all four variants of the 4-choice MCQs, all five variants of the 5-choice MCQs, and all six variants of the 6-choice MCQs, thereby reducing sensitivity to wording and answer-position effects. For MCQs, *accuracy (ACC)* is used as the primary metric. For OEQs, metrics include *Coverage* (whether the answer covers the target clinical relation), *ROUGE* (comparison with the reference rationale), and *BERTScore*.

D Topic Analysis

We also extract question "topics" and analyze their distribution. Topics are generated during the template generation step (see Section 3.3 and Appendix C.3). Each topic represents the primary clinical condition that a question targets, i.e., a "diagnosis" in structured EHR data. After extracting topics from EHRBench, each topic is linked to an ICD-10-CM code, truncated to the 3-digit level (three characters), and used to aggregate the number of questions (Count) and data source share (Percent) together with ICD-10-CM category descriptions. In total, 3,808 unique ICD codes are mapped, and 752 unique codes remain after 3-digit truncation, indicating broad coverage across ICD-10-CM categories and diverse clinical relations in EHRBench. Table 4 summarizes the top-20 ICD-10-CM 3-character categories in the full data source.

Table 4: Top-20 most frequent ICD-10-CM categories in EHRBench. We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
I50	67486	7.11	Heart failure
E87	58869	6.20	Other disorders of fluid, electrolyte and acid-base balance
K59	49915	5.26	Other functional disorders of intestine
I10	43732	4.61	Essential (primary) hypertension
N17	30485	3.21	Kidney failure, acute
I48	26060	2.75	Atrial fibrillation and flutter
I49	22835	2.41	Other cardiac arrhythmias
J96	20986	2.21	Respiratory failure, not elsewhere classified
N18	20966	2.21	Chronic kidney diseases
E86	18484	1.95	Volume depletion
F41	18013	1.90	Other anxiety disorders
F32	17910	1.89	Depressive episode
E78	16597	1.75	Disorder of lipoprotein metabolism and other lipidemias
I20	13811	1.45	Angina pectoris
E03	12535	1.32	Other hypothyroidism
K76	11721	1.23	Other diseases of liver
T81	11167	1.18	Complications of procedures, not elsewhere classified
G89	11001	1.16	Pain, not elsewhere classified
J44	10854	1.14	Other chronic obstructive pulmonary disease
D64	10310	1.09	Other anemia

From Table 4, the most frequent ICD-10-CM 3-character categories correspond to clinically common conditions and acute decompensation syndromes that routinely drive EHR-based decision making, which supports that the EHRBench construction pipeline yields clinically sensible topics and code mappings. In particular, the head of the distribution is dominated by high-prevalence cardiometabolic and acute-care presentations, such as heart failure (I50, 7.11%), fluid/electrolyte and acid–base disorders (E87, 6.20%), functional intestinal disorders (K59, 5.26%), and essential (primary) hypertension (I10, 4.61%). Failure-state phenotypes are also prominent, including acute kidney failure (N17, 3.21%), chronic kidney disease (N18, 2.21%), and respiratory failure (J96, 2.21%), consistent with the fact that many inpatient encounters center on physiologic instability and organ dysfunction. Meanwhile, the presence of comorbidity- and symptom-burden categories—such as volume depletion (E86, 1.95%), pain (G89, 1.16%), anemia (D64, 1.09%), and mental health conditions (F41/F32, 1.90%/1.89%)—suggests that the extracted topics cover both primary diagnoses and common accompanying problems reflected in longitudinal records. At the same time, the distribution exhibits a long-tail pattern: the top-20 ICD-10-CM 3-character categories cover 51% of all questions, implying that nearly half of the benchmark is distributed across a wide range of lower-frequency categories. This long tail is desirable for evaluation because it reduces over-reliance on a small set of frequent conditions and encourages models to generalize beyond the most common clinical scenarios.

Table 5: Top-20 most frequent ICD-10-CM categories in EHRBench (from MIMIC-III). We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
I50	31464	9.85	Heart failure
E87	25992	8.14	Other disorders of fluid, electrolyte and acid-base balance
J96	15236	4.77	Respiratory failure, not elsewhere classified
N17	13565	4.25	Kidney failure, acute
I48	10837	3.39	Atrial fibrillation and flutter
K59	10459	3.27	Other functional disorders of intestine
T81	8989	2.81	Complications of procedures, not elsewhere classified
I10	7967	2.49	Essential (primary) hypertension
N18	6926	2.17	Chronic kidney diseases
E86	5857	1.83	Volume depletion
I49	5320	1.67	Other cardiac arrhythmias
K76	4699	1.47	Other diseases of liver
G40	4696	1.47	Epilepsy and recurrent seizures
I20	4672	1.46	Angina pectoris
E83	4612	1.44	Disorder of mineral metabolism
E03	4145	1.30	Other hypothyroidism
A41	4008	1.25	Other sepsis
E08–E13	3761	1.18	Diabetes mellitus
D64	3716	1.16	Other anemia
J44	3672	1.15	Other chronic obstructive pulmonary disease

We also analyze question topics from each data source (MIMIC-III, MIMIC-IV, and PROMOTE). Tables 5–7 report the top-20 ICD-10-CM categories (truncated to the ICD-10-CM 3-character level) within each data source, including aggregated counts, data source shares, and code descriptions. Across all three sources, the head of the distribution is broadly consistent with the full-dataset pattern: common cardiometabolic and decompensation-related categories repeatedly appear among the most frequent topics, such as heart failure (I50), essential (primary) hypertension (I10), electrolyte and acid–base disorders (E87), functional intestinal disorders (K59), acute and chronic kidney disease (N17/N18), atrial fibrillation and related arrhythmias (I48/I49), and angina pectoris (I20). This cross-source agreement suggests that the extracted topics capture clinically prevalent conditions that are routinely documented in structured EHRs, and further supports that the EHRBench topic extraction and ICD mapping procedures are clinically sensible rather than being driven by data source-specific artifacts. At the same time, each data source exhibits distinct secondary emphases consistent with its underlying cohort and documentation patterns, including higher prevalence of respiratory failure (J96) and sepsis (A41) in MIMIC-III, symptom-oriented categories such as nausea/vomiting (R11) and abdominal pain (R10) in MIMIC-IV, and additional cardiovascular and peri-procedural complications (e.g., I63, I95, I51, T87) in PROMOTE.

We also analyze question topics separately for the three clinical decision tasks (diagnosis, treatment, and prognosis). Tables 8–10 summarize the top-20 ICD-10-CM categories (ICD-10-CM 3-character level) within each task subset. Overall, the three subsets share a consistent head dominated by common cardiometabolic and decompensation-related conditions, such as heart failure (I50), fluid/electrolyte and acid–base disorders (E87), essential (primary) hypertension (I10), atrial fibrillation and related arrhythmias (I48/I49), acute kidney failure (N17), chronic kidney disease (N18), and respiratory failure (J96). These patterns indicate that the extracted topics reflect clinically plausible task-specific distributions while remaining broadly consistent across tasks.

Table 6: Top-20 most frequent ICD-10-CM categories in EHRBench (from MIMIC-IV). We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
K59	17895	6.97	Other functional disorders of intestine
E87	13242	5.16	Other disorders of fluid, electrolyte and acid-base balance
I10	10482	4.08	Essential (primary) hypertension
E86	10210	3.98	Volume depletion
I50	9186	3.58	Heart failure
N17	8837	3.44	Kidney failure, acute
F32	8666	3.37	Depressive episode
F41	8533	3.32	Other anxiety disorders
E78	6624	2.58	Disorder of lipoprotein metabolism and other lipidemias
E03	6321	2.46	Other hypothyroidism
N18	5754	2.24	Chronic kidney diseases
I49	5152	2.01	Other cardiac arrhythmias
G89	4935	1.92	Pain, not elsewhere classified
I48	4858	1.89	Atrial fibrillation and flutter
K76	4246	1.65	Other diseases of liver
J44	3996	1.56	Other chronic obstructive pulmonary disease
D64	3850	1.50	Other anemia
R11	3290	1.28	Nausea and vomiting
I20	2712	1.06	Angina pectoris
R10	2244	0.87	Abdominal and pelvic pain

Table 7: Top-20 most frequent ICD-10-CM categories in EHRBench (from PROMOTE). We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
I50	26836	7.20	Heart failure
I10	25283	6.79	Essential (primary) hypertension
K59	21561	5.79	Other functional disorders of intestine
E87	19635	5.27	Other disorders of fluid, electrolyte and acid-base balance
I49	12363	3.32	Other cardiac arrhythmias
I48	10365	2.78	Atrial fibrillation and flutter
N18	8286	2.22	Chronic kidney diseases
N17	8083	2.17	Kidney failure, acute
E78	8005	2.15	Disorder of lipoprotein metabolism and other lipidemias
F32	6469	1.74	Depressive episode
I20	6427	1.72	Angina pectoris
F41	6275	1.68	Other anxiety disorders
M62	5653	1.52	Other disorders of muscle
I63	5409	1.45	Cerebral infarction
I95	5392	1.45	Hypotension
E55	5253	1.41	Vitamin D deficiency
R11	4676	1.25	Nausea and vomiting
I51	4321	1.16	Complications and ill-defined descriptions of heart disease
T87	3985	1.07	Complications peculiar to reattachment and amputation
K21	3764	1.01	Gastro-esophageal reflux disease

Table 8: Top-20 most frequent ICD-10-CM categories in EHRBench (from the diagnosis subset). We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
I50	19914	8.08	Heart failure
E87	13627	5.53	Other disorders of fluid, electrolyte and acid-base balance
I49	13062	5.30	Other cardiac arrhythmias
N17	9880	4.01	Kidney failure, acute
I10	8584	3.48	Essential (primary) hypertension
I48	8165	3.31	Atrial fibrillation and flutter
J96	6427	2.61	Respiratory failure, not elsewhere classified
N18	6284	2.55	Chronic kidney diseases
E78	5632	2.29	Disorder of lipoprotein metabolism and other lipidemias
D64	5449	2.21	Other anemia
F41	5321	2.16	Other anxiety disorders
I51	4554	1.85	Complications and ill-defined descriptions of heart disease
M62	3869	1.57	Other disorders of muscle
E86	3620	1.47	Volume depletion
R53	3485	1.41	Malaise and fatigue
G47	3155	1.28	Sleep disorders
T81	2856	1.16	Complications of procedures, not elsewhere classified
J81	2848	1.16	Pulmonary edema
K76	2781	1.13	Other diseases of liver
R65	2701	1.10	Symptoms and signs specifically associated with systemic inflammation and infection

Table 9: Top-20 most frequent ICD-10-CM categories in EHRBench (from the treatment subset). We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
K59	48763	10.91	Other functional disorders of intestine
E87	36589	8.19	Other disorders of fluid, electrolyte and acid-base balance
I10	32175	7.20	Essential (primary) hypertension
I50	29370	6.57	Heart failure
F32	13329	2.98	Depressive episode
E86	12054	2.70	Volume depletion
I20	11336	2.54	Angina pectoris
I48	9717	2.17	Atrial fibrillation and flutter
G89	9674	2.17	Pain, not elsewhere classified
F41	9236	2.07	Other anxiety disorders
R11	9211	2.06	Nausea and vomiting
E03	8380	1.88	Other hypothyroidism
E78	8224	1.84	Disorder of lipoprotein metabolism and other lipidemias
E08–E13	7921	1.77	Diabetes mellitus
E83	7546	1.69	Disorder of mineral metabolism
J44	6529	1.46	Other chronic obstructive pulmonary disease
A49	6138	1.37	Bacterial infection of unspecified site
G40	5760	1.29	Epilepsy and recurrent seizures
K21	5573	1.25	Gastro-esophageal reflux disease
N17	5131	1.15	Kidney failure, acute

Table 10: Top-20 most frequent ICD-10-CM categories in EHRBench (from the prognosis subset). We report ICD-10 codes truncated to three characters, along with the total number of questions (Count), the corresponding data source share (Percent), and code descriptions.

ICD	Count	Percent (%)	Description
I50	18202	7.12	Heart failure
N17	15474	6.05	Kidney failure, acute
N18	11690	4.57	Chronic kidney diseases
J96	9983	3.90	Respiratory failure, not elsewhere classified
E87	8653	3.38	Other disorders of fluid, electrolyte and acid-base balance
I48	8178	3.20	Atrial fibrillation and flutter
I49	8115	3.17	Other cardiac arrhythmias
T87	6089	2.38	Complications peculiar to reattachment and amputation
T81	5175	2.02	Complications of procedures, not elsewhere classified
I95	4597	1.80	Hypotension
P39	4047	1.58	Other infections specific to the perinatal period
K76	3938	1.54	Other diseases of liver
N95	3879	1.52	Menopausal and other perimenopausal disorders
F41	3456	1.35	Other anxiety disorders
D64	3277	1.28	Other anemia
I10	2973	1.16	Essential (primary) hypertension
E86	2810	1.10	Volume depletion
E78	2741	1.07	Disorder of lipoprotein metabolism and other lipidemias
J44	2686	1.05	Other chronic obstructive pulmonary disease
R00	2664	1.04	Abnormal heart beat

E Extended Contents of the Main Experiment: Benchmarking LLMs on EHRBench Across Clinical Decision Tasks

E.1 Implementation Details

When running experiments, we process questions in batches of ten using instruction prompts to improve efficiency and ensure stable outputs. Inputs are typically around 8,000 tokens, with a maximum context length of 10,240 tokens. To further reduce latency, we enable early stopping once the model produces a complete JSON-formatted answer. All experiments are implemented in Python 3.11.5 and executed on NVIDIA H200 GPUs with CUDA 12.4. For Azure OpenAI Service, we use API version *"2025-03-01-preview"*. We adopt deterministic decoding to ensure reproducibility; for example, when running the open-source general LLM *"llama3-8b"*, we set *"do-sample=False, temperature=0, top-k=1"*. All models are evaluated using the same prompt template.

To ensure a fair comparison across tasks, sources, and choice types with potentially different numbers of generated QA instances, we construct a fixed evaluation subset for every setting: for each source-task-type combination, we take the first 3,000 questions from the first version. Therefore, each LLM is evaluated with 81,000 questions in total, which provides a very comprehensive evaluation.

E.2 Cost Details

We analyze the computational and monetary costs of the main experiment in Section 4.1. Table 11 reports total token usage, end-to-end runtime, API cost (when applicable), throughput defined as tokens per hour, and overall accuracy. Total token usage is tightly clustered around 10–11M tokens across most models, with only a few higher-token runs (e.g., mistral-7b at 12.87M and yi-1.5-34b at 13.04M), indicating that the main comparison is conducted under a largely matched token budget. In contrast, runtime varies widely from 1.01h (smollm3-3b) to 20.87h (llama3.3-70b), so throughput is largely time-determined in this setting. As a result, throughput spans an order of magnitude, from 0.50 M Tokens/h (llama3.3-70b) to 10.21 M Tokens/h (smollm3-3b), suggesting that system-level efficiency differences dominate over token-count differences under the same evaluation protocol. API-based models often achieve higher throughput than similarly sized self-hosted models, but this efficiency comes with non-negligible monetary cost.

The most accurate API-based models are also among the most expensive and slower in practice. For example, gpt-5.2 achieves the highest overall accuracy (70.91%) but incurs the largest API cost ($71.03) and runs at 1.00 M Tokens/h with a 12.17h runtime. The next tier, gpt-4.1 (69.43%) and gpt-5 (69.06%), reduces the cost relative to gpt-5.2 but still requires $40.67–$42.37, with 4.02–6.45h runtime and 1.57–2.52 M Tokens/h throughput. These results indicate a practical trade-off: improvements at the top end of accuracy can require disproportionately larger time and monetary budgets.

To further illustrate this trade-off, Figure 3 plots overall accuracy against throughput. The highest accuracies concentrate at relatively low throughput, and the upper envelope is formed by comparatively slow models in this benchmark setup. Across all models, higher throughput does not imply higher accuracy, and the high-throughput region is primarily populated by weaker models, indicating that efficiency and effectiveness can be decoupled in practice. A mid-throughput band around 4–6 M Tokens/h provides a pragmatic operating point when both quality and runtime matter. For example, mistral-small3-24b reaches 65.01% at 4.51 M Tokens/h, and doctor-r1-8b attains 61.07% at 4.25 M Tokens/h. On the API side, gpt-4.1-nano runs at 5.88 M Tokens/h with 60.48% accuracy, offering materially higher throughput than frontier models but with a noticeable accuracy gap to gpt-4.1 and gpt-5.2. Within model families, scaling typically increases accuracy while reducing throughput. In the Llama series, accuracy rises from 48.90% (llama3-8b) to 63.35% (llama3-70b) and 67.28% (llama3.3-70b), while throughput drops from 4.19 to 1.35 and further to 0.50 M Tokens/h. A similar pattern appears in Qwen: qwen3-4b and qwen3-8b achieve 60.63% and 60.87% at 4.36 and 3.22 M Tokens/h, whereas qwen3-32b improves to 66.78% but slows to 1.31 M Tokens/h. These within-family trends reinforce that accuracy gains from scaling are accompanied by systematically lower throughput under a fixed evaluation protocol.

Table 11: Cost and efficiency of benchmarking LLMs on EHRBench. We report total token usage (Tokens (M)), end-to-end runtime (Time (h)), API cost (Money ($); not applicable to self-hosted models), throughput defined as total tokens divided by total time (Tokens (M)/h), and overall accuracy (Overall Acc (%)↑).

Model	Total Cost			Throughput	Overall
	Tokens (M)	Time (h)	Money ($)	(M Tokens/h)↑	Acc (%)↑
Open source general LLMs					
glm4-9b	10.46	3.40	–	3.08	59.62
glm4-32b	10.42	10.53	–	0.99	66.12
llama3-8b	10.47	2.50	–	4.19	48.90
llama3-70b	10.45	7.73	–	1.35	63.35
llama3.1-8b	10.51	3.20	–	3.28	56.76
llama3.2-3b	10.41	1.27	–	8.19	49.85
llama3.3-70b	10.45	20.87	–	0.50	67.28
mistral-7b	12.87	3.33	–	3.87	38.23
ministral-8b	10.51	2.72	–	3.87	56.48
mistral-small3-24b	10.41	2.31	–	4.51	65.01
qwen2.5-3b	10.54	1.79	–	5.89	37.87
qwen2.5-7b	10.54	2.52	–	4.18	57.74
qwen2.5-32b	10.37	7.14	–	1.45	64.97
qwen3-4b	10.48	2.40	–	4.36	60.63
qwen3-8b	10.54	3.27	–	3.22	60.87
qwen3-32b	10.39	7.92	–	1.31	66.78
smollm3-3b	10.29	1.01	–	10.21	45.82
yi-1.5-9b	12.47	5.88	–	2.12	45.51
yi-1.5-34b	13.04	5.17	–	2.52	58.94
Medical LLMs					
doctor-r1-8b	10.54	2.48	–	4.25	61.07
med42-8b	10.51	1.70	–	6.19	36.48
ultramedical-8b	10.58	2.00	–	5.30	29.02
m1-32b-1k	10.38	7.18	–	1.45	63.21
m1-7b-23k	10.51	2.41	–	4.37	46.08
HIPAA compliant API-based LLMs					
gpt-4.1-nano	10.04	1.71	2.02	5.88	60.48
gpt-4.1-mini	10.14	3.01	7.47	3.37	66.79
gpt-4.1	10.11	6.45	40.67	1.57	69.43
gpt-5-nano	10.15	3.28	1.70	3.10	57.80
gpt-5-mini	10.16	3.71	8.50	2.74	66.12
gpt-5	10.14	4.02	42.37	2.52	69.06
gpt-5.2	12.14	12.17	71.03	1.00	70.91

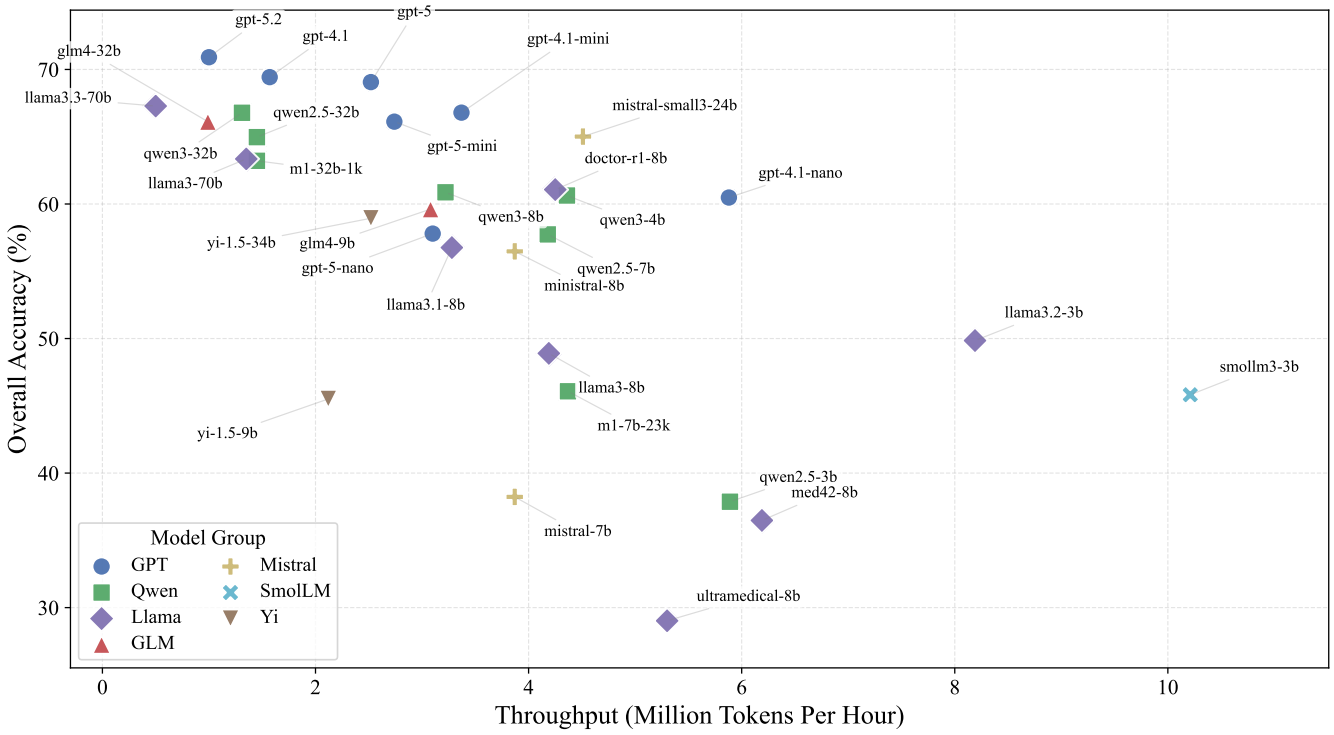

Figure 3: Overall accuracy versus throughput (M Tokens/h) for all evaluated models. Throughput is computed as total tokens divided by end-to-end runtime under the same evaluation protocol. For visualization, medical LLMs are grouped by the family of their corresponding base models: med42, ultramedical, and doctor-r1 are grouped under Llama, while m1 is grouped under Qwen.

E.3 Error Analysis

We analyze errors from all evaluated models and categorize them into three types: (i) prediction errors (Prediction Wrong), which reflect knowledge or reasoning failures under correct formatting; (ii) missing structured outputs (No JSON), where the model does not return a valid JSON object; and (iii) malformed structured outputs (Output Malformed), where the output is not parseable under the required schema. Table 12 summarizes the per-model error breakdown.

Across HIPAA compliant API-based LLMs, format-related failures are essentially eliminated: No JSON is 0.00 and Output Malformed is 0.00–0.01 for gpt-4.1-nano, gpt-4.1-mini, gpt-4.1, gpt-5-nano, gpt-5-mini, gpt-5, and gpt-5.2. Consequently, residual errors are dominated by Prediction Wrong, which ranges from 29.09% (gpt-5.2) to 42.18% (gpt-5-nano), mirroring the accuracy ordering (70.91% to 57.80%). This indicates that, for high-capability API models, the evaluation pipeline is not bottlenecked by response formatting, and measured performance primarily reflects decision quality rather than output compliance.

Most open-source general LLMs show near-perfect compliance (No JSON at 0.00 and Output Malformed close to 0.00), but several models exhibit non-negligible formatting failures that can materially affect end-to-end reliability. For example, mistral-7b shows 18.12% Output Malformed, and qwen2.5-3b shows 13.16%, both coinciding with low accuracies of 38.23% and 37.87%, respectively. A smaller but notable No JSON rate appears in qwen3-8b (1.36%), qwen3-32b (0.44%), and yi-1.5-9b (0.73%), indicating that even strong families can occasionally violate structured-output constraints. In contrast, top open-source performers combine high accuracy with clean outputs, such as llama3.3-70b (67.28% accuracy, 0.00 No JSON, 0.00 Output Malformed) and mistral-small3-24b (65.01%, 0.00, 0.00).

Compared with general-purpose models, medical LLMs show markedly higher rates of structured-output failures, suggesting weaker instruction following under the same JSON-constrained protocol. med42-8b has 25.28% Output Malformed with 36.48% accuracy, and ultramedical-8b has 17.44% Output Malformed and the highest No JSON rate in the table at 1.04%, with 29.02% accuracy. Even the strongest medical model by accuracy, m1-32b-1k (63.21%), exhibits 0.44% Output Malformed, while doctor-r1-8b shows a non-trivial No JSON rate of 0.70% despite 61.07% accuracy. These results imply that, for medical LLMs, reliability issues arise from both decision errors and output-format instability.

Table 12: Error breakdown on EHRBench. We report overall accuracy (Accuracy), missing structured outputs (No JSON), malformed structured outputs (Output Malformed), and prediction errors under valid formatting (Prediction Wrong). All values are percentages (%).

Model	Accuracy (%)↑	No JSON (%)↓	Output Malformed (%)↓	Prediction Wrong (%)↓
Open source general LLMs				
glm4-9b	59.62	0.00	0.01	40.36
glm4-32b	66.12	0.00	0.02	33.86
llama3-8b	48.90	0.00	2.51	48.60
llama3-70b	63.35	0.00	0.02	36.63
llama3.1-8b	56.76	0.00	0.04	43.20
llama3.2-3b	49.85	0.00	0.27	49.88
llama3.3-70b	67.28	0.00	0.00	32.72
mistral-7b	38.23	0.01	18.12	43.63
ministral-8b	56.48	0.00	0.63	42.89
mistral-small3-24b	65.01	0.00	0.00	34.99
qwen2.5-3b	37.87	0.00	13.16	48.97
qwen2.5-7b	57.74	0.00	0.77	41.50
qwen2.5-32b	64.97	0.00	0.10	34.94
qwen3-4b	60.63	0.00	0.00	39.37
qwen3-8b	60.87	1.36	0.05	37.72
qwen3-32b	66.78	0.44	0.04	32.74
smollm3-3b	45.82	0.00	0.57	53.61
yi-1.5-9b	45.51	0.73	2.22	51.54
yi-1.5-34b	58.94	0.00	0.01	41.05
Medical LLMs				
doctor-r1-8b	61.07	0.70	0.04	38.19
med42-8b	36.48	0.20	25.28	38.04
ultramedical-8b	29.02	1.04	17.44	52.50
m1-32b-1k	63.21	0.00	0.44	36.34
m1-7b-23k	46.08	0.37	4.43	49.12
HIPAA compliant API-based LLMs				
gpt-4.1-nano	60.48	0.00	0.01	39.51
gpt-4.1-mini	66.79	0.00	0.00	33.21
gpt-4.1	69.43	0.00	0.01	30.56
gpt-5-nano	57.80	0.00	0.01	42.18
gpt-5-mini	66.12	0.00	0.00	33.88
gpt-5	69.06	0.00	0.00	30.94
gpt-5.2	70.91	0.00	0.00	29.09

E.4 Breakdown Analysis of Medical LLMs versus Base Models

The main experiment in Section 4.1 shows that medical-domain adaptation does not consistently improve performance on EHRBench. To further analyze this phenomenon, we compare medical LLMs with their corresponding base models using question-level aligned evaluation averaged across 4C/5C/6C MCQ variants. Table 13 summarizes the overall and task-specific performance differences.

Three consistent patterns emerge across these comparisons. First, current medical-domain fine-tuning does not reliably improve grounded EHR reasoning performance. This observation further highlights the difficulty of EHR-grounded clinical decision-making, since most existing medical LLM adaptation pipelines are not specifically optimized for reasoning over longitudinal structured EHR contexts. Second, the largest performance degradations are generally observed on Dx tasks, which often require disentangling confounded disease–disease relations and comorbidity patterns. Third, larger medical models narrow the gap relative to their base models, but do not consistently reverse the overall trend.

At the same time, we observe several narrow topic-level exceptions where medical adaptation provides localized gains. For example, m1-32b-1k improves Px performance on topics such as *Angioedema* (+31.60), *Bradycardia* (+18.78), and *Hyperkalemia* (+16.67). m1-7b-23k

Table 13: Comparison between medical LLMs and their corresponding base models on EHRBench. We report overall accuracy and task-specific accuracy differences for diagnosis (Dx), treatment (Tx), and prognosis (Px). Overall results are reported as medical model / base model (Δ), and task-specific columns report the absolute accuracy difference Δ in percentage points.

Pair	Overall Acc (%)	Dx Δ	Tx Δ	Px Δ
med42-8b vs llama3-8b	36.0 / 48.0 (-12.08)	-10.91	-17.83	-8.48
m1-7b-23k vs qwen2.5-7b	45.4 / 56.9 (-11.52)	-17.78	-8.94	-8.37
m1-32b-1k vs qwen2.5-32b	62.3 / 64.2 (-1.92)	-3.53	-1.09	-1.24

shows localized improvements on *Heart failure* Tx (+14.58) and *Obstructive Sleep Apnea* Px (+13.68). med42-8b also improves on several Dx topics, including *Cold intolerance* (+14.71) and *Osteoporosis, unspecified* (+14.58). These exceptions suggest that domain-specific tuning may help in narrow clinical niches, even though it does not consistently improve general EHR-grounded clinical decision-making performance.

Overall, these results suggest that strong performance on EHRBench requires capabilities beyond biomedical terminology familiarity or medical text exposure alone. Models must reason over real longitudinal EHR contexts and resolve clinically confounded relations, including disease progression patterns and disease–treatment associations. Improving these capabilities may require training signals beyond conventional domain adaptation, such as large-scale clinical case supervision and decision-focused objectives. EHR-grounded resources such as EHRBench may therefore provide a useful foundation for future clinical reasoning-oriented model development. Similar observations have also been reported in prior work [19, 20, 109].

E.5 Comparison with Embedding-based non-LLM Baselines

To provide additional reference points beyond LLM-based evaluation, we compare EHRBench with several embedding-based retrieval baselines under the same zero-shot QA setting. For each question, the model encodes the question together with each candidate option and selects the option with the highest cosine similarity. We evaluate these methods on the same 27,000 6C questions used in the main experiment in Section 4.1.

Table 14: Comparison between embedding-based retrieval baselines and LLMs on EHRBench. We report overall accuracy and breakdowns by clinical decision task and data source. Dx, Tx, and Px denote diagnosis, treatment, and prognosis, respectively; MIII, MIV, and PRO denote MIMIC-III, MIMIC-IV, and PROMOTE, respectively. All values are percentages (%).

Model	Overall	Dx	Tx	Px	MIII	MIV	PRO
SapBERT	16.5	15.8	12.7	21.2	17.1	17.0	15.6
SentenceTransformer	27.0	30.3	19.0	31.7	29.0	28.0	23.9
PubMedBERT	32.8	36.9	26.2	35.3	35.2	32.8	30.3
llama3-8b	43.8	39.6	58.4	33.4	45.1	43.4	42.9
qwen3-8b	55.3	52.2	70.4	43.3	56.1	57.1	52.7
gpt-5.2	66.8	68.1	77.3	55.1	67.3	66.9	66.2

Embedding-based retrieval baselines consistently underperform reasoning-capable LLMs across all evaluation settings. Even the strongest biomedical encoder baseline, PubMedBERT, achieves only 32.8% overall accuracy, substantially below general-purpose LLMs such as llama3-8b (43.8%) and qwen3-8b (55.3%). The gap becomes even larger for the strongest API-based model, gpt-5.2 (66.8%).

The largest performance differences are observed on treatment questions. PubMedBERT reaches only 26.2% accuracy on Tx, whereas qwen3-8b and gpt-5.2 achieve 70.4% and 77.3%, respectively. This pattern suggests that many treatment questions in EHRBench cannot be solved through simple semantic similarity or terminology matching alone. Instead, successful prediction often requires reasoning over clinically grounded relations between diagnoses, interventions, and longitudinal patient context.

Overall, these results support the design objective of EHRBench as a benchmark for clinical reasoning rather than retrieval-oriented matching. Strong performance requires models to integrate biomedical knowledge with contextual inference over structured EHR-derived scenarios, which remains challenging for embedding-only retrieval approaches.

E.6 Robustness to QA Generation LLM Choice

In the main construction pipeline of EHRBench, we use HuatuoGPT-o1-8B as the primary source LLM for generating QA templates and question instances. Although this choice provides a consistent generation protocol, it may raise a potential LLM bias concern. To examine this issue and strengthen the robustness and validity of EHRBench, we conduct an additional source-model bias analysis.

Specifically, we regenerate held-out 4C subsets from the same 400 patients using three different medical LLMs as source generators: HuatuoGPT-o1-7B, HuatuoGPT-o1-8B, and m1-7b-23k. Each regenerated subset covers three data sources (MIII/MIV/PRO) and three clinical

decision tasks (Dx/Tx/Px). We then evaluate six strong open-source LLMs on each regenerated subset. This design allows us to test whether the relative ordering of evaluated models remains stable when the QA generation model changes while the underlying patient set, task coverage, and evaluation protocol are held fixed. The results are summarized in Table 15.

Table 15: Performance comparison by changing QA generation LLM choice on EHRBench. Since HuatuoGPT-o1-8B is used as the primary LLM for EHRBench generation in the main construction pipeline, we regenerate held-out 4C subsets from the same 400 patients using three QA generation LLMs and evaluate six open-source LLMs on each subset. All values are accuracy percentages (%).

Eval Model	HuatuoGPT-o1-7B	HuatuoGPT-o1-8B	m1-7b-23k
llama3-70b	61.1	69.1	61.4
llama3-8b	48.4	53.3	43.6
qwen2.5-32b	59.6	66.2	58.8
qwen2.5-7b	57.3	62.4	56.5
qwen3-32b	61.8	70.1	62.4
qwen3-8b	56.1	66.9	56.9

The absolute accuracies vary across source generators, suggesting that different source LLMs can produce QA subsets with different difficulty levels. However, the relative model ordering remains highly stable. Kendall's W across the three source-generator settings is 0.937 ($p = 0.015$), and the pairwise Spearman correlations are 0.829, 0.943, and 0.943. These results indicate that, although the source generator affects the absolute difficulty of the regenerated subset, the main comparative conclusions are robust to the choice of source LLM. Therefore, the observed model rankings in EHRBench are unlikely to be an artifact of relying on HuatuoGPT-o1-8B as a single benchmark-construction model.

E.7 Robustness to Context Event Size

We further evaluate whether the main conclusions are sensitive to the number of context events used to construct the question scenario. In the main construction pipeline of EHRBench, each QA scenario is generated from a compact EHR context consisting of two context events together with the relation-subject entity. This design follows the benchmark objective of evaluating clinical decision-making under partial observation, rather than long-context retrieval over a large number of EHR events.

To assess whether this design choice affects the relative model comparison, we construct an aligned 4C subset with 300 templates across three clinical decision tasks (Dx/Tx/Px) and three data sources (MIII/MIV/PRO). We then vary the number of context events from 2 to 4 to 6 while keeping the templates, answer choices, evaluated models, and inference protocol fixed. The results are shown in Table 16.

Table 16: Performance comparison by changing context event size on EHRBench. In the main construction pipeline, each QA scenario uses two context events together with the relation-subject entity. We evaluate an aligned 4C subset while varying the number of context events from 2 to 4 to 6. All values are accuracy percentages (%).

Model	2 Events	4 Events	6 Events
llama3-70b	45.3	45.0	44.7
llama3-8b	36.3	36.1	35.5
qwen2.5-32b	44.7	44.1	44.6
qwen2.5-7b	41.4	41.4	38.7
qwen3-32b	46.9	46.9	46.7
qwen3-8b	40.5	40.4	37.9

The relative model ordering remains the same across all three context-event settings, suggesting that the main conclusions are not driven by the specific use of two context events in the main benchmark. Larger 32B/70B models are comparatively robust when additional local context is provided, whereas smaller 7B/8B models show slightly greater sensitivity. For example, qwen3-32b remains nearly unchanged from 46.9% with 2 events to 46.7% with 6 events, while qwen3-8b decreases from 40.5% to 37.9%. Overall, these results indicate that increasing the amount of local EHR context does not materially change the model ranking pattern, further supporting the robustness of the main evaluation protocol.

F Additional Experiment: Testing Reasoning LLMs on EHRBench

Recent reasoning-oriented LLMs explicitly produce intermediate reasoning traces during inference, which substantially increases token usage and may interact with context-length constraints. To prevent these token-intensive behaviors from confounding the main benchmarking results and ensure a fair comparison, we report a separate analysis that characterizes accuracy-efficiency trade-offs under different reasoning-effort configurations. All experiments in this section use a maximum context length of 10,240 tokens.

We evaluate ten configurations in total. For gpt-5-nano and gpt-5-mini [81], we test four reasoning-effort settings (minimal/low/medium/high), where minimal matches the setting used in the main experiments. We additionally include gpt-oss-20b and gpt-oss-120b [3], which also enforce explicit reasoning output and therefore incur substantially higher token costs than the standard models evaluated in the main benchmark. To control evaluation cost while maintaining broad coverage, all reasoning models are served through the HIPAA-compliant Azure API, and evaluation is performed on a fixed subset of 1,000 QA items for each source–task–MCQ-type combination. This protocol yields $1{,}000 \times 3 \times 3 \times 3 = 27{,}000$ questions per configuration. Figure 4 summarizes overall accuracy and total token usage across these configurations. Other settings are kept the same as the main experiment.

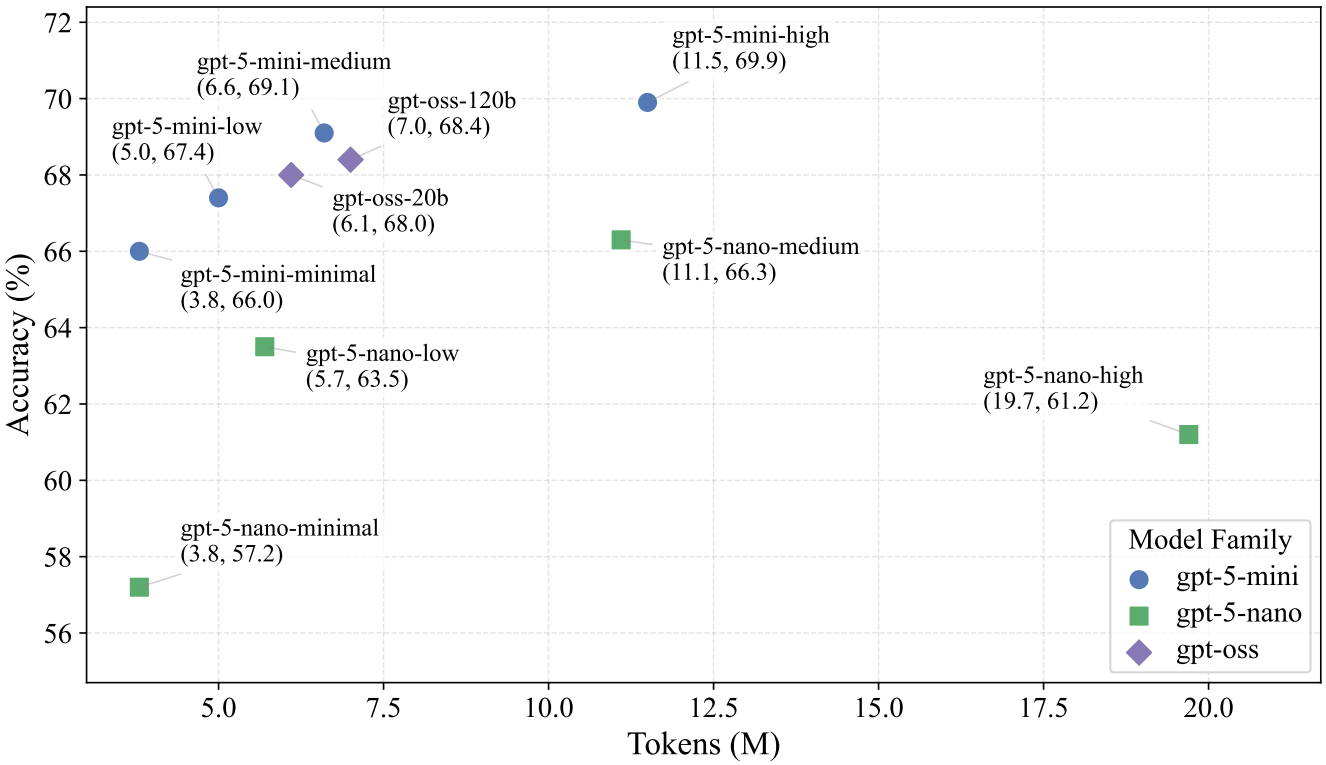

Figure 4: Overall performance and token cost of reasoning model configurations on EHRBench. Each point corresponds to one configuration and is annotated with total token usage (in millions) and overall accuracy.

Figure 4 indicates that higher model capacity and greater reasoning effort generally correspond to higher accuracy and higher token cost, which is consistent with the expected scaling trends and supports the validity of the EHRBench construction pipeline. At matched effort levels, gpt-5-mini outperforms gpt-5-nano, and gpt-oss-120b slightly exceeds gpt-oss-20b. Within gpt-5-mini, increasing effort from minimal to low and then to medium yields monotonic gains. A notable exception occurs for gpt-5-nano: the high-effort setting underperforms the medium setting despite consuming substantially more tokens (19.7M vs. 11.1M). Output inspection indicates that gpt-5-nano-high more frequently fails to return a final decision due to exceeding context or token limits (approximately 10% of cases), which plausibly explains the accuracy degradation at the highest effort level.

The results also exhibit a consistent performance-cost trade-off, as higher reasoning effort incurs diminishing returns. For gpt-5-mini, moving from minimal (3.8, 66.0) to low (5.0, 67.4) and then to medium (6.6, 69.1) yields steady improvements, whereas the step from medium to high increases accuracy by only 0.8 points (69.1→69.9) while increasing token usage by 4.9M (6.6→11.5). This pattern suggests that medium-level reasoning can provide a more efficient operating point when token budgets are constrained.

G Additional Experiment: Testing LLMs with Multiple Versions of Questions from EHRBench

Table 17: Performance on multiple deterministic question versions in EHRBench. Acc (%) denotes mean accuracy; V-Std (pp) denotes the standard deviation of accuracy across versions of the same question; V-Cons. (%) denotes the fraction of questions whose predicted option remains identical across versions of the same question. Overall aggregates results across 4/5/6-choice MCQs.

Model	Overall (%)			4C (%)			5C (%)			6C (%)		
	Acc	V-Std	V-Cons.	Acc	V-Std	V-Cons.	Acc	V-Std	V-Cons.	Acc	V-Std	V-Cons.
glm4-9b	58.63	2.03	81.62	64.11	1.78	84.73	58.28	2.04	81.62	53.49	2.17	78.50
llama3-8b	46.82	3.01	65.18	53.74	2.96	71.11	45.95	3.16	64.53	40.76	2.92	59.89
llama3.2-3b	48.90	2.27	70.80	54.83	2.06	75.36	48.10	2.36	70.34	43.77	2.33	66.70
ministral-8b	55.71	2.22	78.35	61.24	2.51	80.73	55.33	1.85	78.19	50.55	2.30	76.12
qwen2.5-3b	36.88	3.22	60.29	46.16	3.34	67.66	36.07	2.60	59.62	28.41	3.58	53.59
qwen2.5-7b	57.11	1.99	81.38	61.94	1.97	84.11	56.84	1.57	81.32	52.54	2.30	78.71
qwen2.5-32b	64.32	1.64	88.36	69.29	1.67	90.31	63.86	1.52	88.21	59.80	1.72	86.55
qwen3-4b	60.30	1.74	83.93	65.77	1.72	86.97	60.06	1.42	84.13	55.06	1.98	80.68
qwen3-8b	60.90	1.91	84.45	66.57	1.92	87.30	60.51	1.44	84.37	55.61	2.21	81.68
qwen3-32b	65.98	1.73	87.15	70.77	1.87	89.10	65.96	1.29	87.58	61.20	1.94	84.78

To further evaluate robustness to multiple deterministic question versions in EHRBench, we conduct an extended evaluation. Each version paraphrases the clinical context while preserving the same clinical meaning; meanwhile, answer options are systematically permuted so that each option, including the correct answer, appears in each position exactly once across versions. We select ten representative open-source LLMs spanning model scales to support a fair and comprehensive comparison: small models (llama3.2-3b, qwen2.5-3b, qwen3-4b), mid-sized models (glm4-9b, llama3-8b, ministral-8b, qwen2.5-7b, qwen3-8b), and large models (qwen2.5-32b, qwen3-32b). We evaluate the first 1,000 questions for 4-choice, 5-choice, and 6-choice MCQs, with 4/5/6 versions, respectively. In total, each model is evaluated on $1{,}000 \times 15 \times 3 \times 3 = 135{,}000$ questions. Other settings are kept the same as in the main experiment.

We evaluate models using three metrics: accuracy (Acc), variability across versions (V-Std), and prediction consistency across versions (V-Cons.). Let $c \in \{4, 5, 6\}$ denote the choice size, and let $V_c = c$ denote the number of deterministic versions for c-choice questions (paraphrase + answer permutation). For each base question q, the model produces one predicted option $\hat{y}_q^{(v)} \in \{A, \dots\}$ under version $v \in \{1, \dots, V_c\}$.

For each version v, define the version-level accuracy as

$$\mathrm{Acc}^{(v)} = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}\left[\hat{y}_q^{(v)} = y_q\right], \tag{14}$$

where Q is the evaluated set of base questions and y_q is the gold answer. We report Acc as the mean of $\mathrm{Acc}^{(v)}$ over versions.

To quantify robustness to version perturbations, we measure the standard deviation of version-level accuracies:

$$\text{V-Std} = \sqrt{\frac{1}{V_c} \sum_{v=1}^{V_c} \left(\mathrm{Acc}^{(v)} - \overline{\mathrm{Acc}}\right)^2}, \quad \overline{\mathrm{Acc}} = \frac{1}{V_c} \sum_{v=1}^{V_c} \mathrm{Acc}^{(v)}. \tag{15}$$

A smaller V-Std indicates more stable performance across paraphrase/permutation versions under the same choice size.

We further measure whether a model makes the *same* prediction across versions for each base question. Define a per-question consistency indicator:

$$\mathrm{Cons}(q) = \mathbb{I}\left[\hat{y}_q^{(1)} = \hat{y}_q^{(2)} = \cdots = \hat{y}_q^{(V_c)}\right]. \tag{16}$$

The overall consistency is then

$$\text{V-Cons.} = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{Cons}(q). \tag{17}$$

Higher V-Cons. implies that predictions are less sensitive to version perturbations, complementing V-Std, which measures accuracy fluctuation at the aggregate level.

The evaluation results are reported in Table 17. Across all settings, qwen3-32b achieves the highest overall accuracy at 65.98%, followed by qwen2.5-32b at 64.32%. The relative ordering among models is consistent with that in the main experiment, which further supports the correctness of the EHRBench construction pipeline.

Moreover, both V-Std and V-Cons. indicate strong stability across versions, suggesting that evaluation on a single version in the main experiment is reasonable. For example, qwen2.5-32b attains the highest overall consistency (V-Cons.) of 88.36% with the lowest overall variability (V-Std) of 1.64(pp), while qwen3-32b remains close in consistency (87.15) with a V-Std of 1.73. These results indicate that the

evaluated LLMs are robust to deterministic paraphrasing and answer-option permutations, reducing the likelihood that the reported performance is driven by an arbitrary or randomly chosen question version rather than the underlying model capability for CDM.

Increasing the number of answer options consistently reduces accuracy and consistency across all models, and the degradation is substantially larger for weaker models. For the strongest models, the accuracy drops from 4C to 6C is approximately ten percentage points; for example, qwen3-32b decreases from 70.77% (4C) to 61.20% (6C), and qwen2.5-32b decreases from 69.29% to 59.80% (a 9.49-point drop). In contrast, qwen2.5-3b exhibits a much sharper decline from 46.16% to 28.41% (a 17.75-point drop). A similar pattern appears in V-Cons.: qwen2.5-32b drops modestly from 90.31 to 86.55, whereas qwen2.5-3b drops substantially from 67.66 to 53.59. These results suggest that increasing choice cardinality amplifies both difficulty and sensitivity to model capacity, particularly for reliability.

The results reveal a strong qualitative coupling between accuracy and stability: higher-accuracy models generally exhibit lower V-Std and higher V-Cons. For example, qwen2.5-32b and qwen3-32b jointly occupy the top tier in accuracy while maintaining high consistency (above 87) and low variability (at or below 1.73). Nevertheless, the comparison between these two models indicates a nuanced tradeoff: qwen3-32b yields the best overall accuracy (65.98%), whereas qwen2.5-32b achieves slightly higher overall consistency (88.36 vs. 87.15) and the lowest overall V-Std (1.64). This separation between peak accuracy and peak reliability suggests that both metrics are necessary for characterizing clinically relevant robustness, especially under harder settings such as 6C, where both accuracy and consistency decline across all models.

H Additional Experiment: Testing LLMs with Extended Questions from EHRBench

Table 18: Additional evaluation on extended questions from EHRBench across tasks, sources, and question types. We use abbreviations Dx/Tx/Px for diagnosis/treatment/prognosis decision task, MIII/MIV/PRO for MIMIC-III/MIMIC-IV/PROMOTE, and 4C/5C/6C for 4/5/6-choice MCQs. We additionally report evaluation cost in tokens (M) and time (h).

Model	Overall	Task Acc.			Source Acc.			Type Acc.			Cost	
	Acc (%)↑	Dx (%)	Tx (%)	Px (%)	MIII (%)	MIV (%)	PRO (%)	4C (%)	5C (%)	6C (%)	Tokens (M)↓	Time (h)↓
glm4-9b	59.97	58.36	72.26	49.28	62.15	60.42	57.33	65.05	59.74	55.12	34.24	7.14
llama3-8b	48.95	43.46	63.97	39.42	50.20	48.73	47.92	55.48	47.95	43.41	34.31	3.89
ministral-8b	56.86	53.15	71.68	45.75	58.47	57.02	55.08	61.96	56.45	52.16	36.39	4.74
qwen2.5-7b	58.16	56.69	71.56	46.22	59.85	58.86	55.77	62.79	57.91	53.78	34.43	4.53
qwen3-4b	60.95	59.41	73.18	50.24	62.73	60.68	59.43	66.46	60.75	55.63	34.38	5.38
qwen3-8b	61.28	58.36	74.36	51.12	63.05	61.19	59.59	67.14	60.96	55.73	34.64	7.69
llama3.2-3b	49.66	43.07	64.75	41.16	50.34	50.15	48.48	55.33	48.94	44.70	34.38	3.15
qwen2.5-32b	64.95	66.29	76.39	52.18	65.51	65.72	63.62	69.86	64.59	60.41	34.43	23.11
qwen2.5-3b	38.05	35.37	47.80	30.97	39.45	38.99	35.71	47.67	36.69	29.79	34.48	4.17
qwen3-32b	66.42	67.52	76.69	55.04	67.84	66.08	65.33	71.01	66.06	62.17	25.73	25.49

In the main experiment, to ensure efficiency and a fair comparison across tasks, data sources, and question types, we evaluate a fixed subset consisting of the first 3,000 questions for each task–source–type combination. To further assess whether this subset-based protocol faithfully reflects model behavior at scale, we conduct an additional evaluation over the extended question set, which covers all extracted clinical relations in EHRBench. We select ten representative open-source LLMs spanning a wide range of model scales: small models (llama3.2-3b, qwen2.5-3b, qwen3-4b), mid-sized models (glm4-9b, llama3-8b, ministral-8b, qwen2.5-7b, qwen3-8b), and large models (qwen2.5-32b, qwen3-32b). For each model, we evaluate all verified templates of each multiple-choice format (4-choice, 5-choice, and 6-choice), resulting in a total of 180,517 evaluated questions per model. All other settings are identical to those in the main experiment. We report the results in Table 18.

Overall, the extended evaluation yields highly consistent conclusions with the main experiment. In particular, both the overall accuracy and the relative ranking of models closely match those observed under the subset protocol (only 0.15% overall accuracy difference across all models), indicating that the subset-based results are not driven by sampling artifacts or a particular slice of questions. Across the ten representative LLMs spanning small to large scales, the same top-performing models remain at the top and the same weaker models remain at the bottom, with only minor fluctuations in absolute accuracy. This stability suggests that the first 3,000 questions per task–source–type provide sufficient coverage of the underlying relation and question distributions, and that model comparisons are robust to expanding the evaluation set. Consequently, the fixed-subset design offers a practical yet reliable proxy for extended-scale benchmarking, enabling efficient experimentation while preserving the key comparative conclusions about model capability.

Beyond overall performance, the same structural patterns persist across granular slices: treatment remains the easiest task for every evaluated model, whereas prognosis is consistently the most challenging, indicating a stable task-level difficulty imbalance rather than model-specific noise. Source-level differences are also small, suggesting limited sensitivity to data source choice under our evaluation setting. Finally, accuracy monotonically decreases as the number of answer options increases (MCQ-4 > MCQ-5 > MCQ-6), consistent with the main experiment. Taken together, these results further validate that the main experimental design captures the key performance trends and comparative conclusions of extended-scale evaluation on EHRBench.

I Additional Experiment: Testing LLMs with Open-ended Questions from EHRBench

Table 19: Performance on Open-ended questions (OEQs) of EHRBench. We report RC, ROUGE-1, ROUGE-L, and BERTScore (all in %), together with token usage in millions and runtime in hours.

Model	RC (%)	ROUGE-1 (%)	ROUGE-L (%)	BERTScore (%)	Tokens (M)	Time (h)
glm4-9b	28.32	22.98	20.46	42.39	1.21	1.13
llama3-8b	26.93	28.69	24.03	45.82	1.32	1.38
llama3.2-3b	10.87	17.49	15.40	21.20	1.28	0.74
ministral-8b	39.86	28.92	25.60	47.50	1.27	1.13
qwen2.5-3b	4.39	22.85	18.37	40.25	1.23	0.73
qwen2.5-7b	36.09	27.40	24.57	46.32	1.25	0.91
qwen2.5-32b	68.24	34.05	30.82	56.25	1.29	3.79
qwen3-4b	42.89	31.41	27.03	52.08	1.32	1.44
qwen3-8b	61.86	31.41	27.83	52.17	1.31	1.77
qwen3-32b	66.22	30.54	27.75	52.31	1.97	7.20

To further evaluate LLMs on paraphrased open-ended questions (OEQs) in EHRBench, we conduct an extended evaluation using ten representative open-source LLMs spanning different model scales to support a fair and comprehensive comparison, including small models (llama3.2-3b, qwen2.5-3b, qwen3-4b), mid-sized models (glm4-9b, llama3-8b, ministral-8b, qwen2.5-7b, qwen3-8b), and large models (qwen2.5-32b, qwen3-32b). Considering efficiency, we evaluate the first 1,000 OEQs for each source-task setting; in total, each model is evaluated on $1{,}000 \times 3 \times 3 = 9{,}000$ questions. Other settings are kept the same as in the main experiment.

We report four automatic metrics for OEQ evaluation, including RC, ROUGE-1, ROUGE-L, and BERTScore. Specifically, for each OEQ item $I_j = (S_j, Q_j, B_j)$ derived from a template P_k, the model produces a free-text answer $\hat{a}_j$, which is compared against the reference rationale a_j stored in the template. RC measures whether $\hat{a}_j$ covers the target clinical relation $R_k = (x_k, r_k, y_k)$ by checking whether the answer recovers the intended entity (or clinically equivalent surface forms) x_k under concept normalization. ROUGE-1 and ROUGE-L quantify lexical overlap between $\hat{a}_j$ and a_j, where ROUGE-1 emphasizes unigram-level overlap and ROUGE-L captures sequence-level similarity via the longest common subsequence. BERTScore measures semantic similarity between $\hat{a}_j$ and a_j using contextual token embeddings (from the BERT model *bert-base-uncased*) and soft matching, which makes it less sensitive to paraphrasing than ROUGE. For efficiency, we aggregate the total number of (prompt+completion) tokens consumed across all evaluated OEQs and the total wall-clock runtime to obtain Tokens (M) and Time (h), respectively. The results are presented in Table 19.

The results in Table 19 show a clear scale-dependent trend that is consistent with the main experiments: larger models achieve substantially better OEQ quality across all metrics, while small models perform poorly. In particular, qwen2.5-32b achieves the strongest overall quality, reaching 68.24% RC, 34.05% ROUGE-1, 30.82% ROUGE-L, and 56.25% BERTScore, which outperforms all other evaluated models. Notably, qwen3-32b underperforms qwen2.5-32b across all reported metrics, indicating that model scaling alone does not guarantee superior OEQ performance across model families. Overall, these results align with the observations in the main experiments and further support that EHRBench can reliably differentiate the open-ended clinical reasoning capabilities of LLMs at different scales.

OEQs also reveal a strong quality-efficiency trade-off. The fastest runtimes (0.73–0.74h) are achieved by llama3.2-3b and qwen2.5-3b, but both exhibit severe quality loss: llama3.2-3b drops to 10.87% RC and 21.20% BERTScore, while qwen2.5-3b is even lower in RC (4.39%) despite a moderate BERTScore (40.25%). These results indicate that speed alone does not guarantee usable OEQ performance and suggest a clear lower-capacity regime where models generate quickly but fail to meet accuracy-oriented criteria. In contrast, mid-sized models provide more reliable quality with modest cost (e.g., ministral-8b reaches 39.86% RC with 1.13h runtime, and qwen3-8b reaches 61.86% RC with 1.77h runtime). Finally, although qwen3-32b approaches the best RC (66.22% versus 68.24% for qwen2.5-32b), it incurs substantially higher cost (1.97M tokens and 7.20h), which makes it less attractive under runtime constraints. Overall, these findings suggest that OEQs are more demanding and benefit more from stronger models, and that model scale should be selected to balance quality and efficiency.

J LLMs Utilized in EHRBench

In this paper, we leverage more than 30 representative LLMs released between 2023 and 2025. The set of evaluated LLMs is large and up-to-date, supporting meaningful conclusions about current LLM performance trends on EHRBench. Their detailed descriptions and access links are provided below.

- **glm4-9b**: GLM-4 instruction model with 9B parameters, released as a general-purpose bilingual (Chinese–English) LLM for conversational generation and instruction following.
  HuggingFace: https://huggingface.co/zai-org/GLM-4-9B-0414

- **glm4-32b**: Larger GLM-4 instruction model with 32B parameters, providing higher capacity than the 9B variant and typically used when stronger generation quality is desired under similar prompting.
  HuggingFace: https://huggingface.co/zai-org/GLM-4-32B-0414
- **llama3-8b**: Llama 3 instruction model with 8B parameters, a general-purpose open-weight LLM commonly used as an efficient baseline for instruction following and text generation.
  HuggingFace: https://huggingface.co/meta-llama/Meta-Llama-3-8B
- **llama3-70b**: Llama 3 instruction model with 70B parameters, a larger-capacity variant designed to improve performance on knowledge-intensive generation and complex instruction-following workloads.
  HuggingFace: https://huggingface.co/meta-llama/Meta-Llama-3-70B
- **llama3.1-8b**: Llama 3.1 instruction model with 8B parameters, an updated release in the Llama family that is used as a drop-in general-purpose model under the same prompting interface.
  HuggingFace: https://huggingface.co/meta-llama/Meta-Llama-3.1-8B
- **llama3.2-3b**: Llama 3.2 instruction model with 3B parameters, a lightweight variant intended for low-latency or resource-constrained inference while retaining basic instruction-following capabilities.
  HuggingFace: https://huggingface.co/meta-llama/Llama-3.2-3B
- **llama3.3-70b**: Llama 3.3 instruction model with 70B parameters, a later Llama release that maintains the same open-weight instruction interface, and is evaluated here as a high-capacity general-purpose model.
  HuggingFace: https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct
- **mistral-7b**: Mistral instruction model with 7B parameters, widely used as a compact general-purpose baseline that offers strong practical throughput under open-weight deployment.
  HuggingFace: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
- **ministral-8b**: Ministral instruction model with 8B parameters from the Mistral family, evaluated as a mid-sized open-weight model emphasizing practical instruction-following performance.
  HuggingFace: https://huggingface.co/mistralai/Ministral-3-8B-Instruct-2512
- **mistral-small3-24b**: Mistral Small instruction model with 24B parameters, offering a larger open-weight option than the 7B/8B variants and used when higher capacity is beneficial.
  HuggingFace: https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501
- **qwen2.5-3b**: Qwen2.5 model with 3B parameters, a small multilingual checkpoint commonly used for lightweight inference and as a compact baseline within the Qwen family.
  HuggingFace: https://huggingface.co/Qwen/Qwen2.5-3B
- **qwen2.5-7b**: Qwen2.5 model with 7B parameters, a mid-sized multilingual model that supports instruction-style prompting and serves as a standard open-weight baseline.
  HuggingFace: https://huggingface.co/Qwen/Qwen2.5-7B
- **qwen2.5-32b**: Qwen2.5 model with 32B parameters, a higher-capacity multilingual checkpoint typically used for improved response quality and more complex generation tasks relative to smaller Qwen variants.
  HuggingFace: https://huggingface.co/Qwen/Qwen2.5-32B
- **qwen3-4b**: Qwen3 model with 4B parameters, evaluated as a newer-generation multilingual model in the Qwen series under standard instruction prompting.
  HuggingFace: https://huggingface.co/Qwen/Qwen3-4B
- **qwen3-8b**: Qwen3 model with 8B parameters, evaluated as a mid-sized Qwen3 checkpoint that balances capacity and efficiency for multilingual instruction-style generation.
  HuggingFace: https://huggingface.co/Qwen/Qwen3-8B
- **qwen3-32b**: Qwen3 model with 32B parameters, evaluated as a large Qwen3 checkpoint representing a higher-capacity multilingual baseline under the same prompting and decoding setup.
  HuggingFace: https://huggingface.co/Qwen/Qwen3-32B
- **smollm3-3b**: SmolLM3 model with 3B parameters, a lightweight open-weight model used to study low-resource performance and efficiency under the same evaluation protocol.
  HuggingFace: https://huggingface.co/HuggingFaceTB/SmolLM3-3B
- **yi-1.5-9b**: Yi-1.5 bilingual (Chinese–English) model with 9B parameters, used as an additional open-weight general-purpose baseline with strong Chinese/English coverage.
  HuggingFace: https://huggingface.co/01-ai/Yi-1.5-9B
- **yi-1.5-34b**: Yi-1.5 model with 34B parameters, a larger-capacity Yi checkpoint included to compare scaling behavior under the same evaluation pipeline.
  HuggingFace: https://huggingface.co/01-ai/Yi-1.5-34B

- **doctor-r1-8b**: Doctor-R1 medical model released as a domain-focused checkpoint intended for clinical reasoning and instruction-style medical generation, fine-tuned on the Qwen3-8B model.
  HuggingFace: https://huggingface.co/unicornftk/Doctor-R1
- **med42-8b**: Med42 clinical model (8B) fine-tuned on the Llama3-8B model for medical and biomedical language understanding and instruction following.
  HuggingFace: https://huggingface.co/m42-health/Llama3-Med42-8B
- **ultramedical-8b**: UltraMedical instruction-tuned medical model (8B) based on the Llama3-8B model, designed for medical QA-style prompting and clinical instruction following.
  HuggingFace: https://huggingface.co/TsinghuaC3I/Llama-3-8B-UltraMedical
- **m1-7b-23k**: m1 long-context medical model (7B; 23K variant) based on Qwen2.5-7B, included to study the impact of long-context capacity in clinical-style prompting.
  HuggingFace: https://huggingface.co/UCSC-VLAA/m1-7B-23K
- **m1-32b-1k**: m1 long-context medical model (32B; 1K variant) based on Qwen2.5-32B, included to represent a higher-capacity medical checkpoint with an extended context interface.
  HuggingFace: https://huggingface.co/UCSC-VLAA/m1-32B-1K
- **huatuogpt-o1-8b**: HuatuoGPT-o1 medical reasoning model (8B) fine-tuned on Llama3-8B, included as a medical-domain checkpoint with instruction-style interfaces and clinically oriented training.
  HuggingFace: https://huggingface.co/FreedomIntelligence/HuatuoGPT-o1-8B
- **gpt-oss-20b**: Open-weight GPT-OSS model with 20B parameters, included as an additional open-weight baseline with a GPT-style architecture and publicly released weights.
  HuggingFace: https://huggingface.co/openai/gpt-oss-20b
- **gpt-oss-120b**: Open-weight GPT-OSS model (120B) used in our evaluation setup as a large-capacity open-weight baseline.
  Azure OpenAI: https://learn.microsoft.com/en-us/azure/ai-services/openai/
- **gpt-4.1-nano**: Proprietary GPT-4.1 family model accessed via Azure OpenAI (deployment: *gpt-4.1-nano*), included as a low-latency API model under the same prompting and evaluation protocol.
- **gpt-4.1-mini**: Proprietary GPT-4.1 family model accessed via Azure OpenAI (deployment: *gpt-4.1-mini*), included as a cost-efficient API model for instruction-style generation in our evaluation setting.
- **gpt-4.1**: Proprietary GPT-4.1 family model accessed via Azure OpenAI (deployment: *gpt-4.1*), included as a higher-capacity API model for strong general-purpose instruction following and generation.
- **gpt-5-nano**: Proprietary GPT-5 family model accessed via Azure OpenAI (deployment: *gpt-5-nano*), included as a compact API model in the latest available GPT series within our subscription at evaluation time.
- **gpt-5-mini**: Proprietary GPT-5 family model accessed via Azure OpenAI (deployment: *gpt-5-mini*), included as a mid-sized API model representing the same GPT series under a higher-capacity configuration than *gpt-5-nano*.
- **gpt-5**: Proprietary GPT-5 family model accessed via Azure OpenAI (deployment: *gpt-5-chat*), included as a large-sized API model representing the same GPT series under a higher-capacity configuration than *gpt-5-nano*.
- **gpt-5.2**: Proprietary GPT-5 family model accessed via Azure OpenAI (deployment: *gpt-5.2-chat*), included as a large-sized API model representing the same GPT series under a higher-capacity configuration than *gpt-5-nano*.

K Limitations

While EHRBench enables large-scale, reliable EHR-grounded evaluation of LLMs for clinical decision making, it has limitations. First, its construction uses only structured diagnoses, prescriptions, and procedures, excluding informative modalities (e.g., demographics, vital signs, laboratory tests, and imaging). Second, to make KB verification feasible and limit leakage between scenario context and the queried relation, each template uses a small, fixed context window. We focus on encounter-level settings as they are more reliable and grounded, whereas long-range cross-visit relations are weaker and harder to validate. Accordingly, our prognosis task is framed as next-encounter risk prediction rather than calibrated time-to-event forecasting, reflecting uncertain real-world visit timing. Third, although EHRBench contains 960,067 questions, full-set benchmarking is omitted because inference cost and runtime would be prohibitive for many models, making comparisons impractical. We therefore evaluate a capped subset per data source and task for feasible, fair comparisons across diverse models. Finally, KB support trades recall for precision by favoring relations covered by resources (e.g., SemMedDB and UMLS-linked concepts), potentially under-representing rare, emerging, institution-specific, or context-dependent practices. Future work can extend EHRBench by adding modalities, relaxing fixed-context assumptions with leakage-aware controls, supporting reliable multi-visit reasoning, and broadening verification coverage through KBs and evidence sources.

L Prompt Templates

We summarize the prompts used for relation extraction (Table 20), template completion (Table 21), QA generation for MCQ (Table 22) and OEQ (Table 23), and evaluation for MCQs and OEQs (Tables 24 and 25, respectively).

Table 20: Prompt template for task-grounded clinical relation extraction, used in Stage 1 of template generation.

(A) Common Instruction Skeleton

Section	Template Content (polished & condensed)
ROLE	You are a clinical relation extractor and a *strict JSON generator*.
GOAL	Extract a bounded set of **clinically plausible relations** under the task-specific definitions below and output **exactly one** JSON object. These relations will later be used to construct MCQ items: `entity_1` serves as evidence in the question context, and `entity_2` serves as the correct option. You do **not** generate questions here, but you must select relations that are usable for QA construction (for example, informative targets, feasible distractors, and minimal leakage).
INPUT	You will receive structured EHR content listing clinical events for one or two visits. The task-specific block defines: (i) allowable `entity_1/entity_2` pools, (ii) visit scope (intra-visit versus cross-visit), and (iii) allowed relation labels.
ABSOLUTE OUTPUT RULES	• Output **exactly one** valid JSON object enclosed in **one** `json` code block; output **nothing** else. • Top-level keys must include `"raw_relations"` and `"context_events"`; both must be lists. • Each relation item must contain **exactly** the keys: `"entity_1"`, `"relation"`, `"entity_2"`, `"rationale"`. • Type constraints: `entity_1/entity_2/relation` are non-empty strings; `relation` must be from the task-defined label set; `rationale` is **exactly one English sentence** (no line breaks). • `context_events` must contain a fixed number of **plain strings** (no objects) and must obey strict disjointness from all relation endpoints (see the procedure below). • Never output any medical codes; use human-readable event names only. No placeholders, no comments. Stop after the closing code fence.
Context event string extraction (if present)	Use the following priority rules to form each `context_events` string: • Diagnosis: use `event["description"]` only. • Drug/prescription: use `event["drug"]` or `event["description"]`. • Procedure: use `event["description"]`. Each string must be descriptive and human-readable (no timestamps and no codes).
STEP-BY-STEP PROCEDURE	(1) **Build candidate pools**: derive allowable `entity_1` and `entity_2` strictly from the task scope and the input events. (2) **Propose candidate relations**: form directed pairs `entity_1` → `entity_2` with an allowed label. Enforce: no self-loop; verbatim names only; remove duplicates (same endpoints and meaning). Write a one-sentence rationale for each candidate. (3) **Select final `raw_relations`**: prioritize clinical relevance, informativeness (avoid overly trivial or overly common endpoints), and downstream MCQ usability (diversity, low leakage risk, and clean distractor feasibility). Ensure uniqueness and task-scope compliance. (4) **Select `context_events`** from remaining events not used as any relation endpoint, subject to: • **No-overlap rule**: no string overlap with any `entity_1/entity_2` (case-insensitive substring match). • **Stronger unrelatedness rule**: additionally exclude synonyms, abbreviations, spelling variants, and obvious parent/child concepts of any relation endpoint, and avoid same-topic restatements. • Do not reuse the same event object that instantiated any relation endpoint. (5) **Emit JSON only**: output exactly one `json` code block matching the schema. (6) **Silent self-check**: the schema is exact; the bounded relation count is satisfied; no codes; no duplicates; each rationale is one English sentence; context-event disjointness holds.

(B) Strict Output Schema (must match exactly)

```
json
{
"raw_relations":[
{"entity_1":"...", "relation":"...", "entity_2":"...", "rationale":"..."},
. . .
],
"context_events":["...", "..."]
}
```

(C) Task-Specific Block (choose exactly one)

Task	Requirements
Prognosis (Cross-visit)	Extract relations from **prior-visit** events to **next-visit diagnoses/outcomes**. `entity_1` must come from the prior visit; `entity_2` must be a diagnosis/outcome present in the next visit. `context_events` must be chosen **only** from the prior visit. Allowed labels: `cause`, `affect`, `associate_with`. Prefer relations reflecting progression/complications, treatment effects, or latent conditions that become explicit later.
Diagnosis (Same-visit)	Extract diagnosis–diagnosis relations **within the same visit**. Both `entity_1` and `entity_2` must be diagnoses from the current visit. `context_events` must be diagnoses from the current visit (no drugs/procedures). Allowed labels: `cause`, `affect`, `associate_with`. Prefer complication links, shared mechanisms, or strong comorbidity patterns that support identification of an additional diagnosis in the same visit.
Treatment (Same-visit)	Extract treatment-to-diagnosis relations **within the same visit**. `entity_1` must be a prescription or procedure event; `entity_2` must be a diagnosis from the same visit. `context_events` must be diagnoses (no drugs/procedures). Allowed labels: `drug_treat`, `procedure_treat`. Prefer guideline-consistent therapies and procedures that directly manage or treat the diagnosis.

Table 21: Prompt template for template completion, used in Stage 3 of template generation.

DISTRACTOR_NUMBER: `10 (TEN)`

(A) Common Instruction Skeleton

Section	Template Content (polished & condensed)
ROLE	You are a *strict JSON generator* for clinical QA. You must produce **exactly one** valid JSON object and nothing else.
GOAL	Generate **one clinically meaningful MCQ** for **one identified relation**, with **one correct answer** and distractors, grounded **only** in: (i) the provided structured EHR context (task-defined scope), and (ii) the provided relation object (with `relation_id`, rationale, and KB verification evidence).
INPUT	You will receive: (1) **IDENTIFIED RELATION**: `relation_id`, `entity_1`, `entity_2`, `relation`, `rationale`, `semmed_verification`, and definitions for `entity_1`/`entity_2`. (2) **FORBIDDEN TERMS**: real events from the patient record; they **must not appear** in distractors (case-insensitive substring match).
ABSOLUTE OUTPUT RULES	• Output **exactly one** JSON object enclosed in **one** `json` code block. • Output **nothing** outside the code block; **no** placeholders (for example, `<...>` or `{{...}}`) and **no** comments. • The top-level key must be `"base_questions"` and its value must be a list. • Each question item must follow the schema **exactly** (keys and types). • Stop immediately after closing the code block.
STEP-BY-STEP PROCESS	(1) **Read inputs**: interpret `entity_1` as the observed event and `entity_2` as the target consequence/associated outcome per the task definition; use the rationale and KB evidence as grounding. (2) **Build the MCQ**: the stem must reflect the relation and the task scope (same-visit versus cross-visit). (3) **Set answer/topic**: `answer` must exactly equal the correct option string (task-defined; typically `entity_2`, except for the treatment task); `topic` is a concise target label (default: `entity_2`). (4) **Generate distractors**: each distractor should be plausible in general but **not supported or favored** by the sample context and relation evidence, and it must not leak forbidden terms. Prefer three types: `reversed` (wrong direction), `contradicted` (negated by evidence), and `unrelated` (not implied). (5) **Self-validate (mandatory)**: valid JSON; the schema is exact; #distractors = 10; no forbidden-term leakage; **non-nesting** (no synonyms, abbreviations, or parent-child granularity overlaps among choices).

(B) Strict Output Schema (must match exactly)

```
json
{
"base_questions":[
{
"answer":"...",
"topic":"...",
"distractors":[{"entity_name":"...", "type":"reversed"}, . . . ]
}
]
}
```

(C) Task-Specific Block (choose exactly one)

Task	Requirements
Prognosis (Cross-visit)	Predict a **next-visit diagnosis/outcome**. Relation: prior-visit `entity_1` → next-visit `entity_2`. Correct choice: **`entity_2`**. All distractors: **diagnoses/outcomes only** (no drugs/procedures). Stem intent example: "Given prior history, which diagnosis/outcome is most likely at the next visit?"
Diagnosis (Same-visit)	Infer an **additional diagnosis in the same visit**. Relation: same-visit `entity_1` → same-visit `entity_2`. Correct choice: **`entity_2`**. All distractors: **diagnoses only**; must not appear in forbidden terms; enforce non-nesting aggressively.
Treatment (Same-visit)	Select a **treatment/drug/procedure in the same visit**. Relation: treatment `entity_1` → diagnosis `entity_2`. Correct choice: **`entity_1`**. All distractors: **treatments only**; no brand/generic duplicates, abbreviations, or formulation near-duplicates; no forbidden leakage. Topic default: typically `entity_2` (indication/target diagnosis).

Table 22: Prompt template for MCQ QA generation.

(A) Common Paraphrasing Instruction Skeleton

Section	Template Content (polished & condensed)
ROLE	You are a *strict JSON generator* for **paraphrasing** clinical MCQ question stems.
GOAL	Given an input clinical `context` and an ask-only `question`, generate a fixed set of paraphrased versions of the **same** question. This is a **surface-level paraphrase** task: • Do not invent new content or omit any event. • Do not change clinical intent. • Do not introduce emphasis, causality, or interpretation.
INPUT	You will receive: • `context`: one sentence (or two short sentences) that neutrally summarizes all clinical events. • `question` (ask-only): asks what may occur or what will happen (task-defined) and **must not mention any specific event**.
ABSOLUTE OUTPUT RULES	• Output **exactly one** valid JSON object enclosed in **one** `json` code block; output **nothing** else. • The top-level key must be `"question_versions"` and its value must be a list with a fixed length. • Each item must be a JSON object with **exactly** the required keys, and `version` must form a complete consecutive index set (each appears exactly once). • Stop immediately after closing the code fence.
PARAPHRASE RULES (apply to every version)	• **Neutrality**: treat all events equally; no highlighting (for example, "notably" or "especially") and no causal language. • **Anti-abstraction (critical)**: explicitly name each concrete event from the input; do not replace events with high-level summaries (for example, "medical course", "complex presentation", or "respiratory challenges"). • **Linguistic constraints**: fluent clinical English; similar length across versions; prefer two sentences (at most three short sentences); ensure meaningful syntactic and lexical variation while staying close to the original (high lexical overlap). • **No interpretation**: do not add qualifiers (for example, "severe", "suggesting", or "consistent with") and do not create mechanisms or causal chains.
STEP-BY-STEP (mandatory, silent)	(1) Read input `context` and `question`. (2) Produce the required number of versions by paraphrasing the `context` while preserving neutrality and full event coverage, and keep the `question` ask-only (no event leakage). (3) Self-check silently: the count is correct; the intent is unchanged; the schema is valid; the output contains a single JSON code block only.

(B) Strict Output Schema (must match exactly)

```
json
{
"question_versions":[
{"version":1, "context":"...", "question":"..."},
. . .
]
}
```

(C) Task-Specific Block (choose exactly one)

Task	Context framing constraint (paraphrasable; logic fixed)
Prognosis (Prior → Next)	The `context` must be framed as a **prior-visit** summary and must neutrally cover all events from `sample_context` with equal emphasis. Recommended logic: "At the prior visit, the patient's history included `<all events>`." Do not mention any next-visit outcome in the question.
Diagnosis (Same-visit)	The `context` must be framed as a **current-visit** summary and must neutrally cover all events from `sample_context`. Recommended logic: "At the current visit, the patient's diagnoses included `<all events>`." The question remains ask-only and event-free.
Treatment (Same-visit)	The `context` must be framed as a **current-visit** summary, neutrally covering all events from `sample_context` with equal emphasis. Recommended logic: "At the current visit, the patient's diagnoses included `<all events>`." Do not introduce treatment rationale or prioritization.

Table 23: Prompt template for OEQ generation.

(A) Common Reason-Only Instruction Skeleton

Section	Template Content (polished & condensed)
ROLE	You are a clinical summarizer. Produce a concise reason grounded **only** in the provided verified relation fields.
INPUT	You will receive one JSON object containing: `entity_1`, `entity_2`, `relation`, `rationale`, `entity_1_definition`, `entity_2_definition`.
FORBIDDEN	• Do not mention any knowledge source or database name (for example, "SemMed", "evidence", citations, or similar). • Do not introduce new medical facts beyond the provided fields. • Do not explicitly reference that you were given "definitions" or "rationale".
STYLE	• Neutral, concise clinical English. • Length: **2–4 sentences**.
OUTPUT (STRICT)	Output **exactly one** JSON object enclosed in **one** `json` code block; output **nothing** else. The JSON schema must be: `{"reason":"..."}`.

(B) Strict Output Schema (must match exactly)

```
json
{"reason":"..."}
```

(C) Task-Specific Block (choose exactly one)

Task	Guidance (time framing fixed; do not reinterpret)
Prognosis (Prior → Next)	Explain why `entity_1` supports `entity_2` in a cross-visit setting: prior-visit `entity_1` precedes and is linked to next-visit `entity_2`. Use only the provided relation fields and keep the temporal framing unchanged.
Diagnosis (Same-visit)	Explain why `entity_1` supports `entity_2` within a single visit: same-visit `entity_1` is linked to same-visit `entity_2`. Use only the provided relation fields and keep the visit framing unchanged.
Prescription (Same-visit)	Explain why `entity_2` supports a treatment decision involving `entity_1`: diagnoses (`entity_2`) motivate selection of a treatment (`entity_1`) in the same visit. Use only the provided relation fields and keep it neutral and grounded.

Table 24: Prompt template for evaluating MCQs.

(A) Common Evaluation Instruction Skeleton

Section	Template Content (polished & condensed)
INSTRUCTION	You will be given multiple independent multiple-choice medical questions. For **each** question, select the **single best answer** and return the result under the strict JSON output rules below. Do **not** provide explanations, reasoning, or any text outside the JSON output.
OUTPUT REQUIREMENTS (STRICT)	• Output **exactly one** JSON object enclosed in **one** `json` code block; output **nothing** else. • The JSON object must contain the top-level key `"answers"`. • `"answers"` must be a list of **single capital letters**. • The i-th letter corresponds to the i-th question. • Each letter must be one of the options shown for that question.
SELF-CHECK (silent)	Before outputting, silently verify: valid JSON; exactly one code block; English-only output; the answer count equals the question count; each answer is a single valid capital-letter option.

(B) Required Output Schema (must match exactly)

```
json
{
"answers":[
"A", "C", "B", . . .
]
}
```

Table 25: Prompt template for evaluating OEQs.

(A) Common Evaluation Instruction Skeleton

Section	Template Content (polished & condensed)
INSTRUCTION	You will be given multiple independent clinical open-ended questions. For each question, produce two outputs: `answer` (a short event phrase) and `reason` (a concise explanation).
CRITICAL RULES	• Do not invent new facts; use only what the question implies. • `answer` must be a **short phrase** (not a sentence; no extra words). • `reason` must be **1–3 sentences**. • Do not include literal labels such as "Answer:" or "Reason:" inside the strings. • Do not add any extra keys beyond the required schema.
OUTPUT REQUIREMENTS (STRICT)	• Output **exactly one** valid JSON object enclosed in **one** `json` code block; output **nothing** else. • The JSON object must contain **exactly** two top-level keys: `"answers"` (list of strings) and `"reasons"` (list of strings). • The i-th answer and reason correspond to question i. • The two lists must have the same length and equal the number of questions. • Stop immediately after closing the code fence.
SELF-CHECK (silent)	Before outputting, silently verify: one code block; valid JSON; only keys `"answers"` and `"reasons"`; lengths match question count; each answer is a short phrase without sentence-ending punctuation; each reason is 1–3 sentences; no text outside the JSON block.

(B) Required Output Schema (must match exactly)

```
json
{
"answers":["...","...","..."],
"reasons":["...","...","..."]
}
```