FHIR-Hopper: A Neuro-Symbolic Agent for Reasoning over EHRs

Chanwoo Kim [*1,2] **Gunjan Jalori** [*1] **Justin Chen** [1] **Anthony Z. Faranesh** [1] **Dimitrios Antos** [1] **Xin Liu** [1] **A. Ali Heydari** [1]

Abstract

Electronic health records (EHRs), increasingly exchanged via the Fast Healthcare Interoperability Resources (FHIR) standard, encode patient histories as deeply nested, multi-relational bundles that are difficult for large language models (LLMs) to reason over directly. We introduce *FHIR-Hopper*, a neuro-symbolic agentic framework that projects a FHIR bundle into an episodic chronological graph, materializes a budgeted, saliency-ranked linearization as the LLM's initial context, and exposes deterministic graph-traversal tools so the agent can recover details on demand. Across three realistic FHIR clinical benchmarks, FHIR-Hopper attains the highest accuracy across multiple base LLMs *while keeping average input-token usage stable at* $\sim 20k$ *tokens, a typical* $\sim 10\times$ *reduction over the strongest retrieval baseline on long records*. These results suggest that decoupling structure-aware retrieval from neural reasoning is an effective design for clinical question answering over structured EHRs. Our code is available at: https://github.com/Google-Health/consumer-health-research/tree/main/fhir-retrieval-bench/

1 Introduction

While the widespread adoption of the HL7 Fast Healthcare Interoperability Resources (FHIR) standard (Bender & Sartipi, 2013) has improved syntactic interoperability for electronic health records (EHRs), leveraging large language models (LLMs) to reason over these complex records remains inefficient (Makhni et al., 2025; Li et al., 2024; Idrissi-Yaghir et al., 2025). Despite the momentum of frontier LLMs in clinical applications (Saab et al., 2024; Singhal et al., 2023; Shmatko et al., 2025; Heydari et al., 2025), the prevailing heuristic of flattening and concatenating massive JSON- or XML-formatted FHIR bundles into expanded context windows is fundamentally inefficient and suboptimal for two reasons. First, FHIR's system-to-system serialization injects immense structural boilerplate and metadata, exacerbating context overflow and the "lost in the middle" phenomenon (Liu et al., 2024). Second, sequential text processing destroys the inherently multi-relational, directed graph structure of clinical data (e.g., an `Observation` referencing an `Encounter` anchoring a `Condition`). Forcing LLMs to track scattered universally unique identifier (UUID) cross-references in working memory severs the edges between temporally distributed events. Details about the FHIR standard are provided in Section A.1.

Existing mitigation strategies fall short: format conversions (e.g., JSON-to-YAML) fail to restore graph topology, standard retrieval-augmented generation (RAG) struggles with rigid identifiers and temporal dependencies (Cao et al., 2026; Tang & Yang, 2024; Amugongo et al., 2025), and prior agentic systems that directly query FHIR APIs suffer from hallucinated parameters and compounding multi-hop errors (Lee et al., 2025a; Jiang et al., 2025). Due to space constraints, a comprehensive discussion of related work—including tool-using LLM agents, graph-augmented retrieval, and prior EHR systems—is provided in Section A.2.

To address these limitations, we introduce *FHIR-Hopper*, a neuro-symbolic agentic framework for clinical question answering (Figure 1). FHIR-Hopper projects a patient's bundle into an *episodic chronological graph*, resolving UUID references into explicit edges and grouping events into time-ordered episode nodes. From this graph, FHIR-Hopper materializes a token-budgeted, query-conditioned linearization as the LLM's initial context. The agent subsequently invokes deterministic tools (traversing references, filtering temporally, and searching the hidden graph) to recover details on demand. This delegates deterministic lookups to a symbolic engine, freeing the LLM to focus strictly on higher-order reasoning. More specifically, in this work, our contributions are:

- An *episodic chronological graph* representation for FHIR bundles that preserves multi-relational and temporal structure while minimizing token overhead.
- *FHIR-Hopper*, a neuro-symbolic agent combining bud-

[*]Equal contribution [1]Google Research [2]University of Washington. Correspondence to: A. Ali Heydari <aliheydari@google.com>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

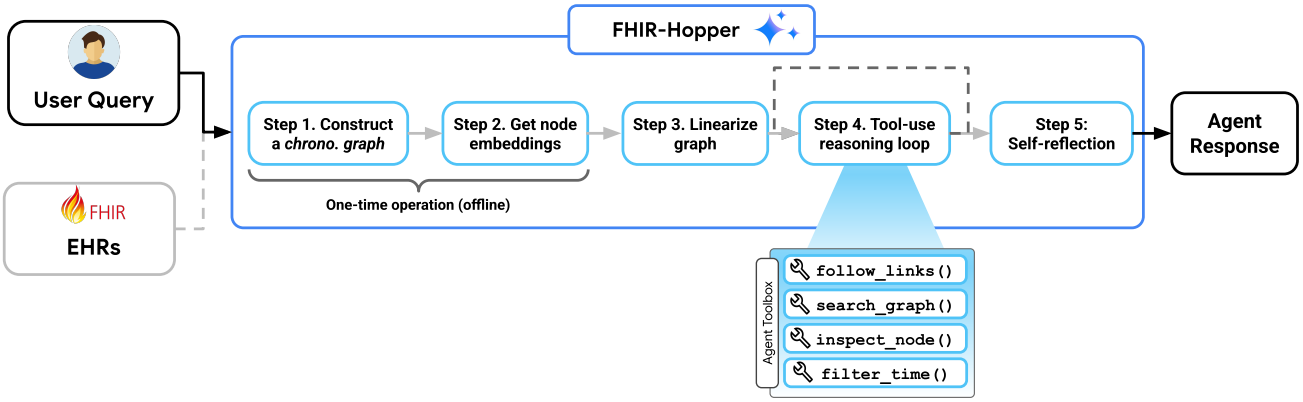

Figure 1. **Overview of the FHIR-Hopper architecture.** The framework projects a raw FHIR bundle into an episodic chronological graph and applies Saliency-Budgeted Linearization (SBL) to produce a token-efficient skeleton view within a specified budget. The LLM agent uses this skeleton together with deterministic tools (`follow_links`, `search_graph`, `inspect_node`, `filter_time`) to interactively traverse the graph and answer complex clinical questions.

geted, saliency-ranked graph linearization with deterministic traversal tools for multi-hop clinical reasoning.

- Empirical evaluations across three FHIR benchmarks and three frontier LLMs, demonstrating that FHIR-Hopper achieves state-of-the-art accuracy while reducing average input-token usage by approximately $10\times$ over the strongest retrieval baseline on long records.

2 Methods

2.1 The FHIR-Hopper Framework

FHIR-Hopper consists of three coupled stages: (1) *episodic graph projection*, (2) *saliency-budgeted linearization* (SBL), and (3) *graph-constrained agentic reasoning*.

Episodic Graph Projection. While FHIR records form a directed graph $G = (V, E)$ of clinical resources V and references E (Tomaszuk et al., 2025a; ALMutairi et al., 2024; Tomaszuk et al., 2025b), this flat view ignores that clinical events cluster into encounters, and reference edges are often sparse. We project the bundle into an *episodic chronological graph* $H = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E} = \{E_1, \ldots, E_T\}$ is a time-ordered set of episode nodes (clinical encounters), and $\mathcal{R}$ contains intra- and inter-episode references (Section A.4). To resolve sparsity, we assign orphan resources (lacking explicit `Encounter` references) via *latent episode inference* (Alg. 1): orphans are attached to the nearest preceding `Encounter` within a temporal window δ, or grouped into a synthetic δ-snapped episode.

Saliency-Budgeted Linearization (SBL). Because H typically exceeds an LLM's token budget β, we construct a token-budgeted linearization L. We first score each resource $v \in V$ against the query q using cosine similarity $\phi(v, q) \in [-1, 1]$ in a shared text-embedding space (Gemini Embedding 1, $d = 3072$ (Lee et al., 2025b)).

Budgeted Selection. Let $x_v \in \{0, 1\}$ indicate if v is in L, and $S = \{v : x_v = 1\}$ be the active set. Let $A(S)$ be the set of *active episodes* containing at least one resource in S. Emitting an active episode incurs a header cost C_{head}, while skipped episodes are compressed into a `[GAP]` marker at cost C_{gap}. The selection objective is:

$$\begin{aligned} \max_{S \subseteq V} \quad & \textstyle\sum_{v \in S} \phi(v, q) \\ \text{s.t.} \quad & |A(S)| C_{\text{head}} + \textstyle\sum_{v \in S} c(v) + N_{\text{gap}}(S) C_{\text{gap}} \leq \beta, \end{aligned} \tag{1}$$

where $c(v)$ is the token cost of v, and $N_{\text{gap}}(S)$ is the number of contiguous inactive episode runs. To showcase performance under extreme constraints, we set $\beta = 4000$, $< 0.1\%$ of average tokens in real-world EHRs (see Section A.3). Since β directly governs the linearized context size and thus token cost, this deliberately small value is a stress test of extreme compression rather than a tuned optimum. β can be scaled up as LLM context windows expand in the future.

Greedy Density Solution. Since $N_{\text{gap}}(S)$ is non-linear, we solve (1) greedily. For a candidate $v \in E_t$, we define its marginal saliency-per-token as:

$$\rho(v \mid S) = \frac{\phi(v, q)}{c(v) + \mathbf{1}[E_t \notin \text{active}(S)]\, C_{\text{head}}} \tag{2}$$

We iteratively add the resource maximizing ρ, maintaining chronological order and emitting gaps between non-adjacent episodes. This concentrates the budget on high-saliency nodes while compressing inactive periods.

Graph-Constrained Agentic Reasoning. The LLM agent receives the SBL view as its *initial* context, which serves as a deliberately cheap, lossy skeleton. To navigate beyond this skeleton and to increase its context, the agent uses four deterministic tools via standard function-calling APIs and finds the final evidence: **1)** `inspect_node(id)`**:**

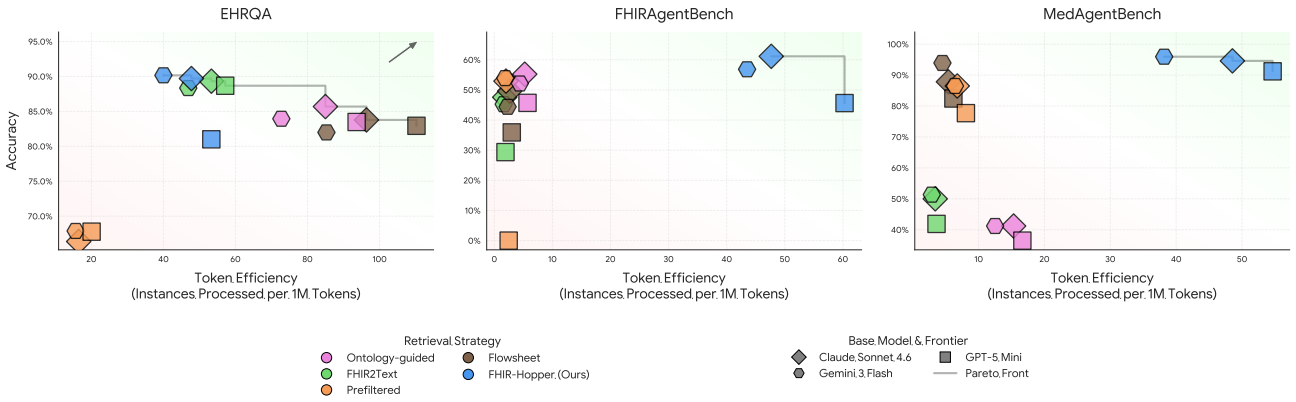

Figure 2. **Pareto efficiency of retrieval strategies.** Axes: accuracy vs. instances processed per 1M tokens (up/right is better, as pointed by the gray arrow in the leftmost plot). FHIR-Hopper strictly dominates the Pareto front on complex records (FHIRAgentBench, MedAgentBench). On short EHRQA records, FHIR-Hopper's lower efficiency stems from its $\mathcal{O}(1)$ fixed overhead (agent prompts/tools). While this overhead outweighs pruning savings on trivial bundles, it avoids the $\mathcal{O}(N)$ context explosion of baselines, enabling scalable reasoning on long, real-world histories.

Returns full JSON of a resource for precise values (e.g., dosages, labs). **2)** `search_graph(keywords)`**:** Semantically searches the hidden graph for pruned nodes. **3)** `follow_links(id)`**:** Traverses explicit relational edges (e.g., `MedicationRequest` → `Condition`) for multi-hop reasoning. **4)** `filter_time(...)`**:** Scopes nodes by date range and type. To prevent runaway loops, we enforce a strict 15-step limit, triggering a graceful termination callback that forces final answer synthesis.

2.2 Datasets

We evaluate on three benchmarks of increasing difficulty (details are provided in Section A.3): **EHRQA**: 5,133 LLM-generated QA pairs over 20 synthetic Synthea bundles (Walonoski et al., 2018) used in Sellergren et al. (2026), testing basic single/two-hop retrieval. **MedAgentBench** (Jiang et al., 2025): 148 clinician-authored retrieval tasks over 100 de-identified patient profiles containing 700K+ clinical data elements. **FHIR-AgentBench** (Lee et al., 2025a): 2,931 clinician-sourced questions on de-identified MIMIC-IV-FHIR records (avg. 9K resources/-patient), requiring complex multi-hop, temporal, and large-bundle reasoning.

2.3 Baselines

We compare FHIR-Hopper against four diverse retrieval strategies: **FHIR2Text**: Heuristically flattens key FHIR resources into natural language sentences and removing all non-text characters.**Ontology-guided** (Kabak et al., 2026): Expands query tokens via a static medical ontology, returning positively scored resources in reverse chronological order alongside fixed anchors (e.g., `Patient`). **Prefiltered** (Schmiedmayer et al., 2025): Filters by clinical heuristics (e.g., keeping only active meds, deduplicating repeat observations) and exposes a condensed identifier triplet (type, name, date) to save tokens. **Flowsheet (SQL on FHIR)** (Grimes et al., 2025): Our implementation of the SOTA tabularization approach, which flattens hierarchical bundles into chronologically ordered Markdown tables (Date, Concept, Value) prepended with patient demographics to support identity-based reasoning.

2.4 Evaluation

We evaluate clinical retrieval performance along two axes: **accuracy** (fraction of correct answers evaluated by an LLM-as-a-judge against deterministic ground truth rules, limiting subjective bias) and **token usage** (total context plus all multi-turn tool-call tokens).

3 Results

Accuracy. FHIR-Hopper consistently attains the highest accuracy across all benchmarks (Table 1, Figure S5). On the simpler benchmark, EHRQA, most baselines exhibit ceiling effects, but FHIR-Hopper alone clears 0.9 (0.902 ± 0.008, Gemini 3 Flash). On the substantially more difficult benchmark, FHIR-AgentBench, FHIR-Hopper (Claude Sonnet 4.6) reaches 0.611 ± 0.018, significantly outperforming all baselines (two-sided McNemar's test, $p < 0.001$). On MedAgentBench, FHIR-Hopper (Gemini 3 Flash) dominates at 0.959 ± 0.030 ($p < 0.01$), while FHIR2Text and Ontology-guided collapse to the 0.4–0.5 range. Crucially, because Ontology-guided uses the exact same Gemini embedding for retrieval, FHIR-Hopper's superiority stems directly from its structural representation and graph-constrained reasoning, rather than embedding quality (ablation detailed in Section A.7).

Context Token Usage. Average input-token usage is reported in Table 1. FHIR2Text and Prefiltered scale poorly on complex records, with FHIR2Text exploding up to 614,787

Table 1. **Accuracy and token usage of FHIR retrieval strategies across base LLMs and benchmarks.** We compare FHIR-Hopper against four baselines. Accuracy is reported as mean ± 95% confidence interval (Acc ↑). Token usage per query is reported as the mean (Tok ↓), where lower is more desirable. The best configuration (highest accuracy, smallest context footprint) on each benchmark is in bold. Note that average token usage could exceed the available context window, since it includes context tokens for all queries. A dash (–) denotes where the model failed to generate responses due to exceeding the context window limit.

Strategy	LLM Backbone	EHRQA		FHIR-AgentBench		MedAgentBench	
		Acc ↑	Tok ↓	Acc ↑	Tok ↓	Acc ↑	Tok ↓
Ontology-guided	Gemini 3 Flash	0.839 ± 0.010	13,748	0.521 ± 0.019	227,060	0.412 ± 0.078	79,949
	Claude Sonnet 4.6	0.857 ± 0.010	11,769	0.552 ± 0.018	190,705	0.412 ± 0.078	65,295
	GPT-5 Mini	0.835 ± 0.010	10,680	0.457 ± 0.018	174,843	0.365 ± 0.081	60,077
FHIR2Text	Gemini 3 Flash	0.883 ± 0.009	21,325	0.453 ± 0.018	614,787	0.514 ± 0.084	345,232
	Claude Sonnet 4.6	0.893 ± 0.008	18,757	0.476 ± 0.018	556,485	0.500 ± 0.084	292,471
	GPT-5 Mini	0.887 ± 0.009	17,498	0.294 ± 0.016	518,598	0.419 ± 0.078	277,005
Prefiltered	Gemini 3 Flash	0.679 ± 0.013	63,978	0.539 ± 0.018	529,171	0.865 ± 0.054	155,893
	Claude Sonnet 4.6	0.664 ± 0.013	60,698	0.529 ± 0.019	496,584	0.865 ± 0.054	147,948
	GPT-5 Mini	0.678 ± 0.013	49,849	–	–	0.777 ± 0.071	123,823
Flowsheet	Gemini 3 Flash	0.819 ± 0.010	11,804	0.446 ± 0.039	418,423	0.939 ± 0.037	221,666
	Claude Sonnet 4.6	0.838 ± 0.010	10,438	0.496 ± 0.017	378,380	0.878 ± 0.051	184,698
	GPT-5 Mini	0.829 ± 0.009	**9,129**	0.361 ± 0.016	331,548	0.824 ± 0.061	161,991
FHIR-Hopper (Ours)	Gemini 3 Flash	$\mathbf{0.902 \pm 0.008}$	24,998	0.568 ± 0.018	22,984	$\mathbf{0.959 \pm 0.030}$	26,196
	Claude Sonnet 4.6	0.897 ± 0.008	20,935	$\mathbf{0.611 \pm 0.018}$	20,978	0.946 ± 0.037	20,603
	GPT-5 Mini	0.810 ± 0.011	18,761	0.456 ± 0.018	**16,589**	0.912 ± 0.044	**18,301**

tokens on FHIR-AgentBench (frequently triggering context overflows). In contrast, FHIR-Hopper maintains a stable footprint of $\sim$20,000 tokens across all benchmarks, an approximate **10× reduction** over the strongest retrieval baseline (Ontology-guided) on FHIR-AgentBench, and up to 30× relative to FHIR2Text.

Ultimately, FHIR-Hopper is strictly pareto-dominant on the longer, real-world benchmarks (FHIR-AgentBench and MedAgentBench), achieving maximum accuracy at the lowest token cost (Figure 2). We explicitly note a deliberate scaling trade-off on the short, synthetic EHRQA bundles, where FHIR-Hopper consumes more tokens than the Flowsheet and Ontology-guided baselines. This occurs because our agentic framework incurs a fixed initialization overhead (tool definitions, system prompts, and episode headers). On trivial records, this fixed overhead outweighs the savings from graph pruning. However, as patient history grows $\mathcal{O}(N)$, as in real-world data, baseline context sizes explode linearly, whereas FHIR-Hopper's budgeted linearization bounds token usage to a stable $\mathcal{O}(1)$ footprint, initialized by the SBL budget, allowing it to scale better to massive clinical histories.

4 Discussion

In this work, we proposed the FHIR-Hopper, an agentic approach that leverages graph representations of FHIR bundles to achieve state-of-the-art performance on clinical EHR question-answering tasks. Our results demonstrate that the FHIR-Hopper not only outperforms other harnesses (FHIR2Text, Ontology-Guided Retrieval, Prefiltered) and serialization baselines in accuracy, but also does so with a significantly reduced context token footprint. The recent momentum in enterprise health AI, e.g. the launch of *GPT Health*, *Perplexity Health*, and *Verily Me*, highlights an unprecedented opportunity to interface LLMs directly with EHRs. However, sustaining this interaction securely and affordably requires moving beyond naive data serialization. FHIR-Hopper provides a scalable blueprint for such integration. By projecting FHIR bundles into navigable graphs, we demonstrate that LLMs can achieve state-of-the-art clinical reasoning while shrinking the context footprint to a stable $\sim$20,000 tokens, even for complex queries (e.g. in FHIR AgentBench).

This token efficiency yields critical operational advantages. **Privacy and Cost:** Transmitting only a budgeted graph skeleton and dynamically retrieved nodes to cloud inference APIs minimizes the exposure of sensitive Protected Health Information (PHI) and drastically cuts token costs. **Auditability:** Unlike monolithic black-box LLMs, FHIR-Hopper produces a deterministic trace of traversed nodes and invoked tools, establishing a transparent reasoning path essential for trust.

While optimized for question answering, this neuro-symbolic architecture naturally extends to **administrative workflows** (automating billing and prior authorizations), **patient-facing tools** (synthesizing longitudinal care plans), **data quality and maintenance** (identifying data inconsis-

tencies and data gaps), and **biomedical research** (structuring EHR cohorts).

Limitations & Trade-offs. First, the multi-turn tool-calling loop adds inference latency: each query requires multiple LLM calls (mean 9.8 on FHIR-AgentBench), while context-stuffing baselines use a single call. Although graph projection and node embeddings are computed offline, the online tool-calling loop increases wall-clock latency in our implementation to roughly 30–60 seconds per query, compared with about 10 seconds for non-agentic approaches. Thus, FHIR-Hopper trades higher latency for improved reasoning performance. Reducing this latency through better agentic design and system-level optimization is an important direction for improving usability in deployment. Second, the framework relies heavily on FHIR reference integrity, and sparse or broken references will degrade graph traversal. Latent episode inference that handles orphan resources and deterministic tools, such as `search_graph`, which retrieves evidence without depending on correct references provide a fallback, partially mitigating this; nonetheless, a prior data curation process may be necessary for severely degraded data. Third, our current implementation focuses on structured tabular data, currently treating other resources such as unstructured clinical notes (`DocumentReference`) and medical imaging as opaque nodes. Our method could be extended by adding tools backed by dedicated encoders for each modality, but we leave this to future work. Also, because clinical scenarios vary in complexity, adapting the linearization budget β per query (e.g., to estimated question difficulty) is a natural extension. In addition, a systematic characterization of failure modes (e.g., hallucinated resource identifiers, temporal-ordering errors, or graph-traversal dead-ends) would be useful for improving the system. Finally, prospective validation on live EHR systems beyond retrospective de-identified records, is important directions for future work.

References

ALMutairi, M., AlKulaib, L., Wang, S., Chen, Z., ALMutairi, Y., Alenazi, T. M., Luther, K., and Lu, C.-T. Fhirviz: Multi-agent platform for fhir visualization to advance healthcare analytics. In *Proceedings of the 15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400713026. doi: 10.1145/3698587.3701392. URL https://doi.org/10.1145/3698587.3701392.

Amugongo, L. M., Mascheroni, P., Brooks, S., Doering, S., and Seidel, J. Retrieval augmented generation for large language models in healthcare: A systematic review. *PLOS Digital Health*, 4(6):e0000877, 2025.

Anthropic. Claude sonnet 4.6 system card, 2026. URL https://anthropic.com/claude-sonnet-4-6-system-card.

Bender, D. and Sartipi, K. Hl7 fhir: An agile and restful approach to healthcare information exchange. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pp. 326–331, 2013. doi: 10.1109/CBMS.2013.6627810.

Cao, L., Chen, Q., and Guo, Y. EHR-RAG: Bridging long-horizon structured electronic health records and large language models via enhanced retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2601.21340.

DeepMind, G. Gemini 3.0 flash preview system card, 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. From local to global: A graph rag approach to query-focused summarization, 2025. URL https://arxiv.org/abs/2404.16130.

Grimes, J., Brush, R., Rhyzhikov, N., Szul, P., Mandel, J., Gottlieb, D., Grieve, G., Sadjad, B., and Sanyal, A. SQL on FHIR - tabular views of FHIR data using FHIRPath. *npj Digital Medicine*, 8(1):342, 2025. doi: 10.1038/s41746-025-01708-w. URL https://doi.org/10.1038/s41746-025-01708-w.

Heydari, A. A., Gu, K., Srinivas, V., Yu, H., Zhang, Z., Zhang, Y., Paruchuri, A., He, Q., Palangi, H., Hammerquist, N., Metwally, A. A., Winslow, B., Kim, Y., Ayush, K., Yang, Y., Narayanswamy, G., Xu, M. A., Garrison, J., Lee, A. A., Vafeiadou, J., Graef, B., Galatzer-Levy, I. R., Schenck, E., Barakat, A., Perez, J., Shreibati, J., Hernandez, J., Faranesh, A. Z., Prieto, J. L., Heneghan, C., Liu, Y., Zhan, J., Malhotra, M., Patel, S., Althoff, T., Liu, X., McDuff, D., and Xu, X. O. The anatomy of a personal health agent, 2025. URL https://arxiv.org/abs/2508.20148.

Idrissi-Yaghir, A., Arzideh, K., Schäfer, H., Eryilmaz, B., Bahn, M., Wen, Y., Borys, K., Hartmann, E., Schmidt, C., Pelka, O., et al. Using a diverse test suite to assess large language models on fast health care interoperability resources knowledge: Comparative analysis. *Journal of Medical Internet Research*, 27:e73540, 2025.

Jiang, J., Zhou, K., Zhao, W. X., Song, Y., Zhu, C., Zhu, H., and Wen, J.-R. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph, 2024. URL https://arxiv.org/abs/2402.11163.

Jiang, Y., Black, K. C., Geng, G., Park, D., Zou, J., Ng, A. Y., and Chen, J. H. Medagentbench: A virtual ehr environment to benchmark medical llm agents. *NEJM AI*, 2(9):AIdbp2500144, 2025. doi: 10.1056/AIdbp2500144. URL https://ai.nejm.org/doi/full/10.1056/AIdbp2500144.

Johnson, A. E. W., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T. J., Hao, S., Moody, B., Gow, B., Lehman, L.-w. H., Celi, L. A., and Mark, R. G. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1, 2023. doi: 10.1038/s41597-022-01899-x. URL https://doi.org/10.1038/s41597-022-01899-x.

Kabak, Y., Erturkmen, G. B. L., Gencturk, M., Namli, T., Sinaci, A. A., Corcoles, R. A., Ballesteros, C. G., Abizanda, P., and Dogac, A. FHIR-RAG-MEDS: Integrating HL7 FHIR with retrieval-augmented large language models for enhanced medical decision support, 2026. URL https://arxiv.org/abs/2509.07706.

Lee, G., Hwang, H., Bae, S., Kwon, Y., Shin, W., Yang, S., Seo, M., Kim, J.-Y., and Choi, E. EHRSQL: A practical text-to-sql benchmark for electronic health records. *Advances in Neural Information Processing Systems*, 35: 15589–15601, 2022.

Lee, G., Bach, E., Yang, E., Pollard, T., Johnson, A., Choi, E., Lee, J. H., et al. Fhir-agentbench: Benchmarking llm agents for realistic interoperable ehr question answering. *arXiv preprint arXiv:2509.19319*, 2025a.

Lee, J., Chen, F., Dua, S., Cer, D., Shanbhogue, M., Naim, I., Ábrego, G. H., Li, Z., Chen, K., Vera, H. S., Ren, X., Zhang, S., Salz, D., Boratko, M., Han, J., Chen, B., Huang, S., Rao, V., Suganthan, P., Han, F., Doumanoglou, A., Gupta, N., Moiseev, F., Yip, C., Jain, A., Baumgartner,

S., Shahi, S., Gomez, F. P., Mariserla, S., Choi, M., Shah, P., Goenka, S., Chen, K., Xia, Y., Chen, K., Duddu, S. M. K., Chen, Y., Walker, T., Zhou, W., Ghiya, R., Gleicher, Z., Gill, K., Dong, Z., Seyedhosseini, M., Sung, Y., Hoffmann, R., and Duerig, T. Gemini embedding: Generalizable embeddings from gemini, 2025b. URL https://arxiv.org/abs/2503.07891.

Li, Y., Wang, H., Yerebakan, H. Z., Shinagawa, Y., and Luo, Y. FHIR-GPT enhances health interoperability with large language models. *NEJM AI*, 1(8):AIcs2300301, 2024. doi: 10.1056/AIcs2300301. URL https://ai.nejm.org/doi/full/10.1056/AIcs2300301.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Makhni, S., Cerrato, P., Rico, J., Niazi, S., O'Horo, J., Peters, S., Shah, V., and Halamka, J. Meeting the challenges of electronic health record (ehr) optimization. *NPJ Digit Med*, 9(1):8, Dec 2025. ISSN 2398-6352 (Electronic); 2398-6352 (Linking). doi: 10.1038/s41746-025-02178-w.

OpenAI. Openai gpt-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

Saab, K., Tu, T., Weng, W.-H., Tanno, R., Stutz, D., Wulczyn, E., Zhang, F., Strother, T., Park, C., Vedadi, E., Chaves, J. Z., Hu, S.-Y., Schaekermann, M., Kamath, A., Cheng, Y., Barrett, D. G. T., Cheung, C., Mustafa, B., Palepu, A., McDuff, D., Hou, L., Golany, T., Liu, L., baptiste Alayrac, J., Houlsby, N., Tomasev, N., Freyberg, J., Lau, C., Kemp, J., Lai, J., Azizi, S., Kanada, K., Man, S., Kulkarni, K., Sun, R., Shakeri, S., He, L., Caine, B., Webson, A., Latysheva, N., Johnson, M., Mansfield, P., Lu, J., Rivlin, E., Anderson, J., Green, B., Wong, R., Krause, J., Shlens, J., Dominowska, E., Eslami, S. M. A., Chou, K., Cui, C., Vinyals, O., Kavukcuoglu, K., Manyika, J., Dean, J., Hassabis, D., Matias, Y., Webster, D., Barral, J., Corrado, G., Semturs, C., Mahdavi, S. S., Gottweis, J., Karthikesalingam, A., and Natarajan, V. Capabilities of gemini models in medicine, 2024. URL https://arxiv.org/abs/2404.18416.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

Schmiedmayer, P., Rao, A., Zagar, P., Aalami, L., Ravi, V., Zahedivash, A., han Yao, D., Fereydooni, A., and Aalami, O. Llmonfhir. *JACC: Advances*, 4 (6_Part_1):101780, 2025. doi: 10.1016/j.jacadv.2025.101780. URL https://www.jacc.org/doi/abs/10.1016/j.jacadv.2025.101780.

Sellergren, A., Kazemzadeh, S., Jaroensri, T., Kiraly, A., Traverse, M., Kohlberger, T., Xu, S., Jamil, F., Hughes, C., Lau, C., Chen, J., Mahvar, F., Yatziv, L., Chen, T., Sterling, B., Baby, S. A., Baby, S. M., Lai, J., Schmidgall, S., Yang, L., Chen, K., Bjornsson, P., Reddy, S., Brush, R., Philbrick, K., Asiedu, M., Mezerreg, I., Hu, H., Yang, H., Tiwari, R., Jansen, S., Singh, P., Liu, Y., Azizi, S., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Riviere, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Buchatskaya, E., Alayrac, J.-B., Lepikhin, D., Feinberg, V., Borgeaud, S., Andreev, A., Hardin, C., Dadashi, R., Hussenot, L., Joulin, A., Bachem, O., Matias, Y., Chou, K., Hassidim, A., Goel, K., Farabet, C., Barral, J., Warkentin, T., Shlens, J., Fleet, D., Cotruta, V., Sanseviero, O., Martins, G., Kirk, P., Rao, A., Shetty, S., Steiner, D. F., Kirmizibayrak, C., Pilgrim, R., Golden, D., and Yang, L. Medgemma technical report, 2026. URL https://arxiv.org/abs/2507.05201.

Shmatko, A., Jung, A. W., Gaurav, K., Brunak, S., Mortensen, L. H., Birney, E., Fitzgerald, T., and Gerstung, M. Learning the natural history of human disease with generative transformers. *Nature*, 647(8088):248–256, 2025. doi: 10.1038/s41586-025-09529-3. URL https://doi.org/10.1038/s41586-025-09529-3.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., Agüera y Arcas, B., Webster, D., Corrado, G. S., Matias, Y., Chou, K., Gottweis, J., Tomasev, N., Liu, Y., Rajkomar, A., Barral, J., Semturs, C., Karthikesalingam, A., and Natarajan, V. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2. URL https://doi.org/10.1038/s41586-023-06291-2.

Tang, Y. and Yang, Y. Multihop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*, 2024.

Tomaszuk, D., Smajevic, A., Sagi, T., and Hose, K. Fhir lens: A graph-based approach to semantic ehr exploration. In *2025 IEEE 38th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 1–6, 2025a. doi: 10.1109/CBMS65348.2025.00133.

Tomaszuk, D., Smajevic, A., Sagi, T., and Hose, K. Fhir lens: A graph-based approach to semantic ehr explo-

ration. In *2025 IEEE 38th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 1–6, 2025b. doi: 10.1109/CBMS65348.2025.00133.

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., and McLachlan, S. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 03 2018. ISSN 1527-974X. doi: 10.1093/jamia/ocx079. URL https://doi.org/10.1093/jamia/ocx079.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023. URL https://arxiv.org/abs/2308.08155.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

A Appendix

A.1 Fast Healthcare Interoperability Resources (FHIR)

Established by HL7 International in 2011, the Fast Healthcare Interoperability Resources (FHIR) framework serves as the preeminent standard for healthcare data storage and exchange. The schema is predicated on several core components:

- **`Resources`**: Modular primitives representing distinct clinical entities (e.g., `Patient`, `Condition`, `Encounter`, `Medication`, `Observation`).
- **`References`**: Directional edges that map relationships from a source resource to a target resource, establishing the overarching graph topology.
- **`Patient`**: The root node for most clinical trajectories, encapsulating administrative and demographic metadata (e.g., identifiers, gender, birth date).
- **`Encounter`**: Represents a specific patient-provider interaction, establishing the spatiotemporal context (e.g., inpatient admission, emergency visit) that anchors subsequent clinical events.
- **`Condition`**: Documents diagnoses, health concerns, or problems, tracking the longitudinal severity and clinical status (e.g., active, relapse, remission) of a patient's ailments.
- **`Observation`**: The primary mechanism for recording measurements and assertions, spanning vital signs, laboratory assays, and social history.
- **`Medication`**: Represents a specific pharmacological substance or mixture, typically instantiated contextually via linked `MedicationRequest` or `MedicationAdministration` resources.

Simplified Patient FHIR bundle

```
[
  {
    "resourceType": "Patient",
    "id": "0a8eebfd-...",
    "gender": "female",
    "birthDate": "2128-05-06",
    "extension": [
      // ... Nested custom demographic attributes (e.g., Race, Ethnicity)
      { "url": ".../us-core-race", "valueString": "White" }
    ]
  },
  {
    "resourceType": "Condition",
    "id": "b9ea8d38-...",
    "subject": { "reference": "Patient/0a8eebfd-..." }, // Relational link to
        Patient
    "encounter": { "reference": "Encounter/6c68c032-..." },
    "code": {
      "coding": [{
        "system": ".../mimic-diagnosis-icd9", // Standardized medical ontology
        "code": "78959",
        "display": "Other ascites"
      }]
    }
  },
  {
    "resourceType": "Encounter",
    "id": "18f74cab-...",
    "subject": { "reference": "Patient/0a8eebfd-..." },
    "period": {
      "start": "2180-07-23T14:00:00-04:00",
      "end": "2180-07-23T23:50:47-04:00"
    }
  }
  // ... {3,607 resources}
]
```

Figure S1. **Simplified Patient FHIR bundle.**

Unlike flat tabular architectures, FHIR structures a patient's longitudinal history as a directed graph of JSON objects interconnected via `References` (Figures S1 and S2). Consequently, `Resources` are deeply nested; reconstructing a coherent clinical sequence typically requires relational joins and temporal grouping by `Encounter`. Further specifications are detailed in the official FHIR documentation (https://www.hl7.org/fhir/).

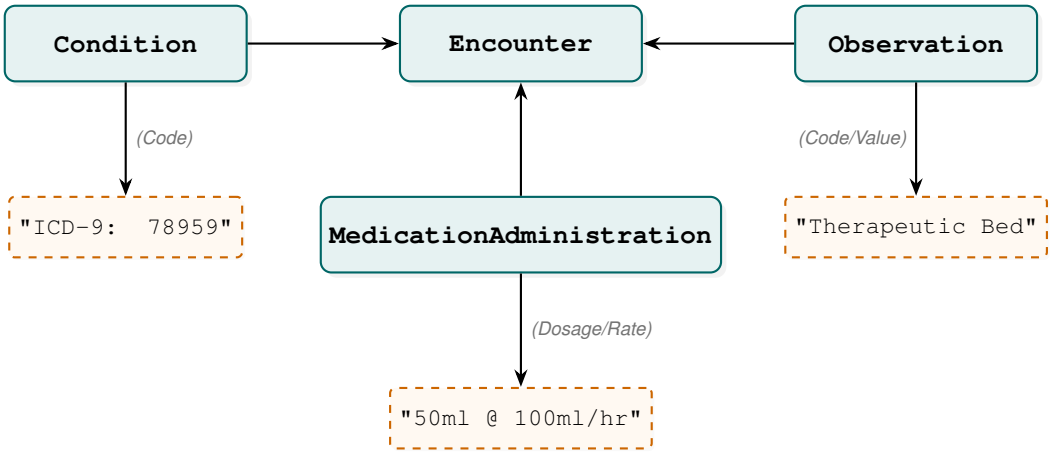

Figure S2. **Graph structure of a simplified FHIR bundle.** Resources such as `Condition`, `Observation`, and `MedicationAdministration` reference their parent `Encounter`, forming a directed graph.

A.2 Related Work

Tool-using LLM agents. A line of recent work equips LLMs with external tools and lets them interleave reasoning with tool calls, including ReAct (Yao et al., 2023), Toolformer (Schick et al., 2023), AutoGen (Wu et al., 2023). Similar to these systems, FHIR-Hopper exposes deterministic tools and uses standard function-calling APIs, but the tools are specialized to graph traversal over a structured clinical representation rather than general web or code execution.

Graph and structure-augmented retrieval. GraphRAG (Edge et al., 2025) and related approaches (Jiang et al., 2024) build a knowledge graph from a corpus and let an LLM traverse or condition on it during retrieval. Our approach is closest in spirit: we treat the FHIR bundle itself as a typed, time-stamped graph and provide both a budgeted linearization and traversal tools. The differences are domain-specific: edges and node types come from the FHIR schema rather than open-domain extraction, and the graph is anchored on an episodic spine rather than a flat similarity graph.

LLMs over FHIR and EHR. A number of systems consume FHIR data with LLMs. LLMs-On-FHIR (Schmiedmayer et al., 2025) relies on heuristic prefiltering and de-duplication; FHIR-RAG-Meds (Kabak et al., 2026) performs ontology-guided retrieval; FHIR-AgentBench (Lee et al., 2025a) and MedAgentBench (Jiang et al., 2025) construct agentic benchmarks in which LLMs call FHIR APIs directly. These systems serve as our baselines or evaluation targets. Other clinical LLMs such as Med-PaLM (Singhal et al., 2023) and Med-Gemini (Saab et al., 2024) are trained or instruction-tuned on clinical text, but do not specifically address structured-resource reasoning over FHIR bundles.

A.3 Datasets

A.3.1 MIMIC-IV FHIR Demo (from FHIR AgentBench)

This benchmark contains 2,931 clinician-sourced questions and answers, originally from the EHRSQL (Lee et al., 2022) dataset grounded in de-identified MIMIC-IV (Johnson et al., 2023) FHIR patient records. Each patient record contains roughly 9K FHIR resources on average, and some questions require reasoning over more than 2,000 relevant resources. MIMIC-IV is a publicly available database from the Beth Israel Deaconess Medical Center covering emergency department and ICU admissions between 2008-2019. The MIMIC-IV Clinical Database Demo on FHIR is a subset of these data that was converted into FHIR R4 specifications for 100 randomly selected patients (94 of which are present in FHIR AgentBench). FHIR resources included patient, organization, observations from specimen samples, medications, charted observations, and billing. Free-text clinical notes were excluded.

The queries in FHIR AgentBench are focused on single patient natural language questions. The benchmark tests agents'

ability to perform multi-step retrieval and reasoning over nested FHIR data, including temporal logic, cross-resource reference traversal, case-sensitive terminology handling, and queries with no relevant matches. We show the token count distribution of the FHIR bundles present in this dataset in Figure S3.

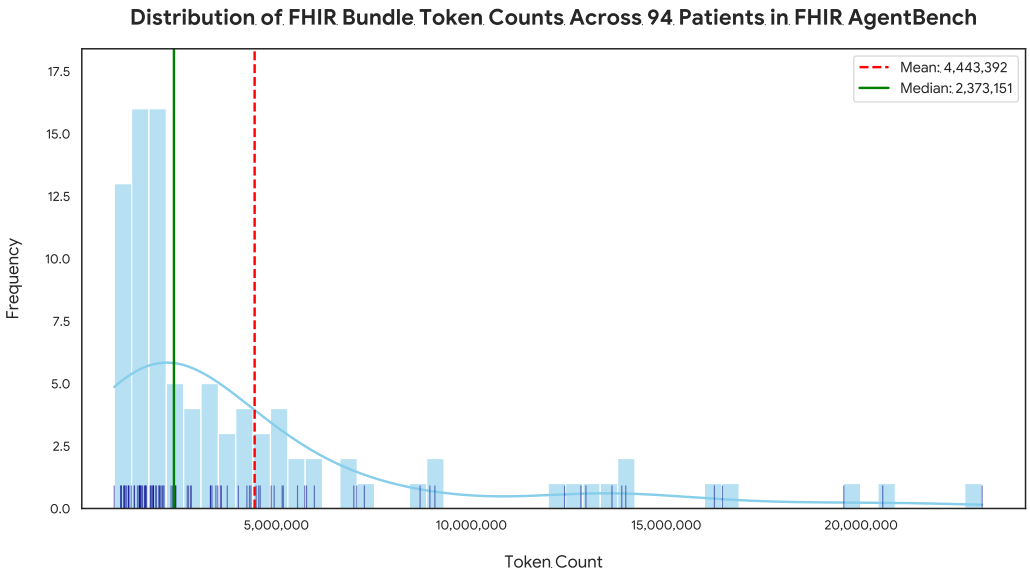

Figure S3. Distribution of Tokens (computed through Gemini's tokenizer) for bundles in the FHIR AgentBench.

A.3.2 MEDAGENTBENCH

MedAgentBench (Jiang et al., 2025) consists of 300 clinically-relevant and verifiable tasks from 10 categories written by licensed human clinicians. The profiles represent 100 patients with over 700,000 individual clinical data elements, including laboratory results, vital signs, procedures, diagnoses, and medication orders. The profiles are derived from real, de-identified patient records with demographic identifiers synthetically generated. Given the scope of our work, we subset the dataset to include *all retrieval* queries (i.e. only patient information retrieval, laboratory result retrieval, and patient data aggregation; "POST" tasks such as *Test ordering*, *Medication ordering*, *Referral ordering*, and *Recording patient data* were excluded), resulting in 148 tasks. We present the token count distribution of the FHIR bundles, for the retrieval queries, in Figure S4.

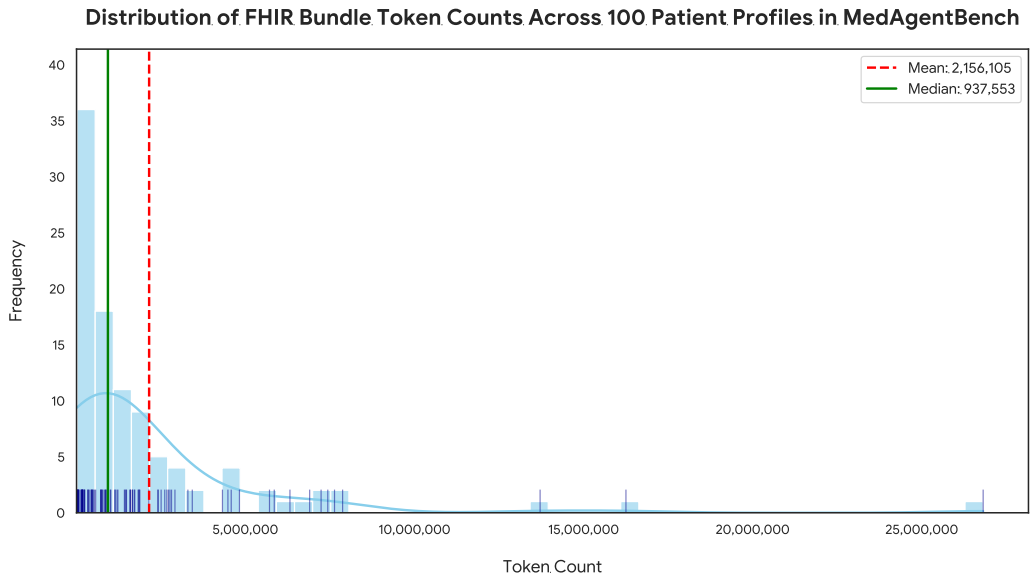

Figure S4. Distribution of Tokens (computed through Gemini's tokenizer) for bundles in the MedAgentBench dataset.

A.3.3 EHRQA

We curated this dataset for granular testing and validation on a simple benchmark. This dataset contains 5,133 LLM-generated multiple-choice question-answer pairs for 20 synthetic FHIR bundles produced by Synthea (Walonoski et al., 2018), which were generated following the process described in Sellergren et al. (2026). All synthetic bundles fit within a 32k context window, and clinical notes were not included in EHRQA. We provide an example of an entry in this dataset in below.

EHRQA Example

```
QAExample(
patient_id='78a7ab19-bdd0-2238-e4e5-de757ee074d1', question='What past nasal
    sinus condition does the patient have?\n',
answer_choices=['(A) acute empyema of nasal sinus', '(B) fungal sinusitis', '(C)
     chronic sinusitis', '(D) viral sinusitis'],
correct_answer='D', ehr_fact=ConditionExistenceInHistoryFact(fact_type=<
    ConditionFactType.EXISTENCE_IN_HISTORY: 'existence_in_history'>,
fact_variation_type=<ConditionFactVariationType.FINDING_SITE: 'finding_site'>,
evidence=[],
evidence_refs=['Condition/6ba0a993-8f90-f024-dc46-89243cb67d17'],
code='2095001',
code_type=<MedicalCodeset.SNOMED: 'http://snomed.info/sct'>, is_transitive=False
    ,
intermediate_codes=()))
```

Because both the bundles and the questions are synthetic, EHRQA exercises only basic single- or two-hop retrieval and exhibits ceiling effects on competent baselines; we report it primarily to verify that strategies do not regress on simpler tasks. Moreover, we wanted to show that while some approaches may achieve promising results on such synthetic benchmarks, they may exhibit significant performance drop once they are put to test in real world setting, as evident by performance results on FHIR AgentBench and MedAgentBench.

A.4 FHIR-Hopper Implementation Details

A.4.1 Pseudocode for episodic graph projection

Algorithm 1 Episodic Graph Projection & Latent Episode Inference

Require: Resource bundle $\mathcal{B}$, timestamp function $\tau(v)$, encounter-reference function ref(v), temporal window δ
Ensure: Chronological episodic spine $\mathcal{E}_{spine}$

```
H_explicit ← ∅ {Map of Encounter ID → explicit episode nodes}
H_latent ← ∅ {Map of δ-snapped timestamp → latent episode nodes}
O ← ∅ {Queue of non-Encounter resources}
// Phase 1: initialize backbone from explicit Encounters
for r ∈ B do
    if r.type == 'Encounter' then
        Initialize episode node E with spine resource r
        H_explicit[r.id] ← E
    else
        O.enqueue(r)
    end if
end for
// Phase 2: assign orphans, falling back to latent episodes
for v ∈ O do
    id_ref ← ref(v) {Encounter reference if present}
    if id_ref ≠ null ∧ id_ref ∈ H_explicit then
        H_explicit[id_ref].add(v)
    else
        t ← τ(v)
        if t == null then continue
        E* ← nearest preceding Encounter with start time in [t − δ, t]
        if E* ≠ null then
            H_explicit[E*.id].add(v)
        else
            t' ← snap t to a grid of width δ
            if t' ∉ H_latent then
                Create new latent episode E_latent anchored at t'
                H_latent[t'] ← E_latent
            end if
            H_latent[t'].add(v)
        end if
    end if
end for
// Phase 3: chronological merge
E_union ← Values(H_explicit) ∪ Values(H_latent)
E_spine ← SortByTime(E_union) {Chronologically ordered episodic spine}
return E_spine
```

A.4.2 SYSTEM PROMPT

System Prompt

```
# Role
You are a Clinical Graph Navigator operating on a chronological hypergraph of
    patient data. Providing medical advice or a diagnosis is outside the scope
    of your role.

# Objective:
Your goal is to provide your best answer to the user's query by navigating this
    graph and utilizing the tools at your disposal. Note that you are a query
    system, not a chat model, and should not converse with the user by asking
    follow-up questions.


### 1. THE CONTEXTUAL CONSTRAINT
You are viewing a **Budget-Constrained Saliency Map** (the Skeleton).
* **Around 99% of the data is HIDDEN** to fit in your context window.
* **Relevance Scores** (`[0.92]`) show semantic similarity to your query.
* **Gaps** (`[... GAP ...]`) indicate skipped timeframes with low relevance.

You will be provided with a budget-constrained Saliency Map and a user query.
    Make sure to consider the user query and the
"BUDGETED SKELETON (TOP HITS)" that will show you the top hits in the graph that
     are most relevant to the query.
However, it is likely that this representation may be missing some data that is
    relevant to the query. Therefore, it is crucial for you to carefully
assess whether the information you see in the SKELETON is SUFFICIENT for you to
    answer the query, and also check the HIDDEN graph
to see if there are any additional clues or data that may be relevant to best
    answer the user's query.

### 2. BUDGET WARNING (CRITICAL)
You have a strict maximum budget of **{{ max_llm_calls }} steps**. Each step can
     consist of multiple tool calls.
You must gather all necessary information and provide your final answer before
    exceeding this limit.
Plan your tool calls efficiently.

### 3. TEMPORAL AWARENESS (CRITICAL)
You should take into account the current time when answering questions including
     relative time specifiers, like "last month". For example, "last month"
    means the 30 days prior to current time. You can use the `filter_time` tool
    to query this specific date range.

### 4. CONTEXTUAL INFORMATION
You are provided with the following contextual information for your query:

{{ context_str }}

### 5. YOUR TOOLS
You must navigate this graph dynamically to find the truth.

1. `inspect_node(resource_id)`: Fetches the full JSON content of a specific FHIR
     resource.
   * **Usage:** "Read the file." Fetches the full JSON content.
   * **Mandatory:** You CANNOT verify values (BP, Dates, Dosage) without this.
   * **Args:**
      * `resource_id`: A specific ID from the skeleton (e.g., "Observation/123").
```

```
2. `search_graph(keywords)`: Searches the FULL (hidden) graph for resources
    containing keywords.
   * **Usage:** "Search the dark." Scans the HIDDEN portion of the graph.
   * **When to use:** The Skeleton is missing a key node.
   * **Args:**
      * `keywords`: Clinical keywords (e.g., "Troponin", "Discharge Summary").
          Also use this to find atemporal data like Patient demographics,
          Location details, etc.

3. `follow_links(resource_id)`: Return the IDs and Types of resources directly
    referenced by the node.
   * **Usage:** "Traverse the graph based on references in resources." Returns
       the IDs of resources *referenced by* the target node.
   * **When to use:** You have a node (e.g., MedicationRequest) and need to find
        its reason (reasonReference -> Condition) or its author (requester ->
       Practitioner), but those linked nodes are not visible in the current
       skeleton.
   * **Args:**
      * `resource_id`: The ID of the node you want to trace FROM.

4. `filter_time(start_date, end_date, resource_type, keywords)`: Filters graph
    nodes by time range, resource type, and keywords.
   * **Usage:** "Find all observations between 2023-01-01 and 2023-12-31"
   * **When to use:** The skeleton does not have the chronological details you
       need, and you want to specifically query a time period. Crucial when the
       user specifies a timeline ("since August 2180" or "last two weeks").
       Remember to frame the boundaries relative to simulation's end date: {{
       max_timestamp }}.
   * **Args:**
      * `start_date`: ISO format YYYY-MM-DD
      * `end_date`: ISO format YYYY-MM-DD
      * `resource_type`: The FHIR resource type to filter by (e.g., "Observation
          ")
      * `keywords`: Keywords to search for in the node data

Note that THERE IS NO `finish` tool. You should NOT hallucinate on calling a `
    finish` tool.

Make sure to also refer to the docstring that will be provided to you for each
    tool to double check usage and arguments that you need to pass to each tool
    in case things have changed.

### 6. NAVIGATION STRATEGY
Follow this priority queue to solve the query:

* **PHASE 1: SCAN (The Skeleton)**
   * Look for High-Relevance (`[0.9+]`) nodes in the Skeleton.
   * *Constraint:* If the question is about Medications, it may not be good to
       trust the skeleton's absence. Medications often have low semantic
       similarity to condition queries. Move to Phase 2.

* **PHASE 2: EXPAND (The Neighborhood)**
   * If you found a relevant node (e.g., a "Diabetes" Diagnosis) but lack
       details, use `follow_links` on it.

* **PHASE 3: HUNT (The Void)**
   * If the Skeleton and Links are empty, use `search_graph` or `filter_time`.

* **PHASE 4: VERIFY (The Grounding)**
   * **CRITICAL:** Never guess a value. You must `inspect_node` to see the
       actual JSON numbers before answering.
```

A.5 Judge for Comparing Model Response against Ground Truth

Because models generate free-form text, exact-match grading is overly rigid. Since the tasks are verifiable with the provided ground truth, we use an LLM-as-a-judge for a simple correctness assessment: given the query, the ground truth, and the model response, the judge returns a correctness verdict and a short rationale. Because the QA tasks resolve to deterministic ground truth (specific values, dates, codes), the judge's room for subjective interpretation is bounded. As a small validation study, we used humans to verify our auto-rater on a randomly selected subset of the FHIR-AgentBench ($n = 291$, $\sim 10\%$ of all queries), and found the LLM judge to not make any mistakes in assessing answer correctness.

Judge Prompt

```
You are a clinical auditor. Evaluate the Model Response against the Ground Truth.

Return a JSON object with two fields:
1. "is_correct": boolean (true if the model answer accurately answer the
   question that aligns with the ground truth)
2. "reason": string (brief explanation)

**Patient Query**: {user_query}
**Ground Truth Answer**: {ground_truth}
**Model Response**: {model_response}

JSON Output:
```

A.6 Model Hyperparameters

To assess generalizability, we evaluate every method against three frontier LLMs spanning different providers and cost/latency profiles:

- **Gemini 3 Flash** (DeepMind, 2025): a highly efficient model developed by Google DeepMind, optimized for speed and cost, with a large context window suitable for long clinical records.
- **Claude Sonnet 4.6** (Anthropic, 2026): a model from Anthropic offering strong reasoning capability with moderate inference cost.
- **GPT-5 Mini** (OpenAI, 2025): a lightweight OpenAI model with a smaller context window, used to evaluate robustness under tighter resource constraints.

All models use default top-p and a max output of 8,192 tokens. We set temperature to 0.6 for Gemini 3 Flash and Claude Sonnet 4.6; GPT-5 Mini uses its API-default 1.0. FHIR-Hopper uses the same temperature as its backbone LLM. Models are accessed via Google Vertex AI (Gemini 3 Flash, Claude Sonnet 4.6) and the OpenAI API (GPT-5 Mini). Saliency scoring uses the Gemini Embedding 1 model ($d = 3{,}072$) consistently across configurations to isolate the effect of the backbone LLM from the embedding. The judge is Claude Sonnet 4.6 with temperature 0.0 and max output $8{,}192$ tokens.

A.7 Ablation Studies

Table S1. **Ablation study.** We ablate FHIR-Hopper by removing tool access (W/O TOOLS), forcing the agent to rely solely on the SBL view of the graph.

STRATEGY	BASE MODEL	EHRQA	FHIR-AGENTBENCH	MEDAGENTBENCH
FHIR-HOPPER	GEMINI 3 FLASH	**0.902 ± 0.008**	0.568 ± 0.018	**0.959 ± 0.030**
	CLAUDE SONNET 4.6	0.897 ± 0.009	**0.611 ± 0.018**	0.946 ± 0.037
	GPT-5 MINI	0.810 ± 0.011	0.456 ± 0.018	0.912 ± 0.044
W/O TOOLS	GEMINI 3 FLASH	0.844 ± 0.010	0.595 ± 0.017	0.784 ± 0.064
	CLAUDE SONNET 4.6	0.859 ± 0.010	0.591 ± 0.018	0.784 ± 0.064
	GPT-5 MINI	0.838 ± 0.010	0.233 ± 0.015	0.473 ± 0.084

We ablate FHIR-Hopper by removing tool access, forcing the agent to rely on the SBL view alone (Table S1). Tools are most valuable on EHRQA and MedAgentBench (e.g., MedAgentBench with GPT-5 Mini drops from 0.912 to 0.473) and for the weaker GPT-5 Mini backbone, where tools partially compensate for weaker base reasoning. On FHIR-AgentBench the effect is small and not in a single direction, suggesting the SBL view alone already surfaces most of the evidence needed there.

A.8 Additional Results

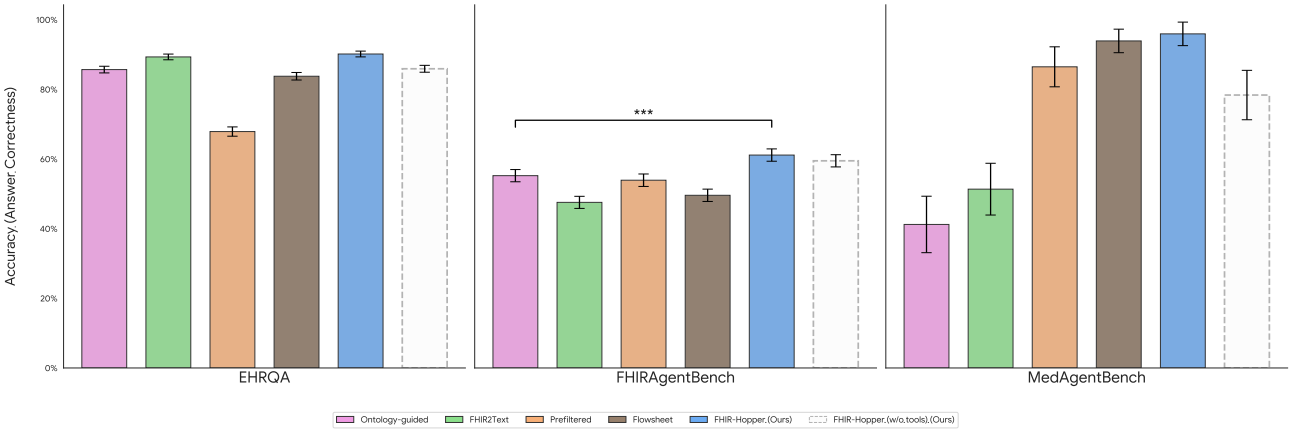

Figure S5. **Accuracy of FHIR retrieval strategies across benchmarks.** We compare FHIR-Hopper against the baseline strategies on EHRQA, FHIR-AgentBench, and MedAgentBench; for each strategy we report the configuration with the highest-performing backbone LLM. FHIR-Hopper attains the highest accuracy on every benchmark. Asterisks denote statistical significance ($p < 0.001$, McNemar's test) of the improvement over the second-best strategy that is not ours.

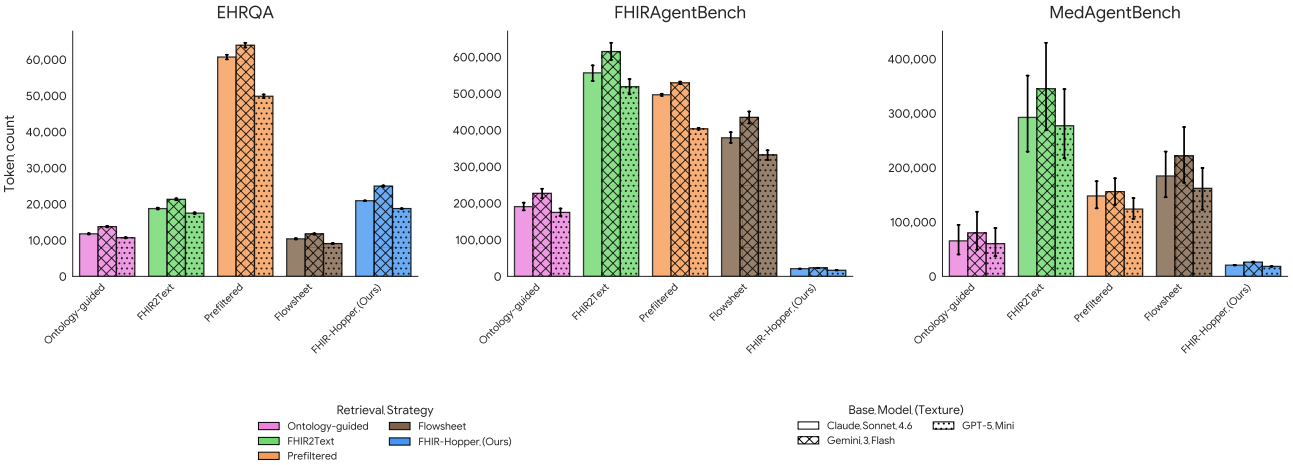

Figure S6. **Token usage per strategy and LLM (lower is better).** We report the average number of tokens consumed per query with the three LLMs used. FHIR-Hopper maintains a stable token footprint across all benchmarks, particularly on FHIR-AgentBench where baselines exceed context-window limits. Error bars indicate 95% confidence intervals.

A.9 Broader Impact and Ethics

Auditability: Unlike black-box LLMs, our agent produces a trace of nodes visited. This is crucial for clinical validation and trust. Additionally, by sending only relevant graph nodes to the LLM (rather than the full record), we minimize the data footprint exposed to cloud inference API.

The deployment of autonomous agents in clinical settings carries significant ethical, safety, and transparency implications. Foremost among these is the risk of hallucinations, which could lead to severe patient harm if left unchecked. Therefore, systems like the FHIR-Hopper must be deployed strictly as assistive tools rather than autonomous decision-makers. Rigorous human-in-the-loop verification remains essential to ensure that AI-generated insights are clinically sound, safe, and transparent before influencing patient care.