HypEHR: Hyperbolic Modeling of Electronic Health Records for Efficient Question Answering*

Yuyu Liu[1] **Sarang Rajendra Patil**[2] **Mengjia Xu**[2] **Tengfei Ma**[3]
[1]Department of Computer Science, Stony Brook University
[2]Department of Data Science, New Jersey Institute of Technology
[3]Department of Biomedical Informatics, Stony Brook University
{yuyu.liu, tengfei.ma}@stonybrook.edu
{sp3463, mx6}@njit.edu

Abstract

Electronic health record (EHR) question answering is often handled by LLM-based pipelines that are costly to deploy and do not explicitly leverage the hierarchical structure of clinical data. Motivated by evidence that medical ontologies and patient trajectories exhibit hyperbolic geometry, we propose HypEHR, a compact Lorentzian model that embeds codes, visits, and questions in hyperbolic space and answers queries via geometry-consistent cross-attention with type-specific pointer heads. HypEHR is pretrained with next-visit diagnosis prediction and hierarchy-aware regularization to align representations with the ICD ontology. On two MIMIC-IV-based EHR-QA benchmarks, HypEHR approaches LLM-based methods while using far fewer parameters. Our code is publicly available at https://github.com/yuyuliu11037/HypEHR.

1 Introduction

Electronic health record (EHR) question answering (EHR-QA) aims to answer natural-language clinical questions over a patient's longitudinal record (Bardhan et al., 2023). For example, "has patient been admitted to the emergency room on the first hospital visit" or "is there any microbiological test result on the current hospital visit for patient's blood culture?" (Bae et al., 2023). Recent datasets over MIMIC-III/IV have driven progress, but most methods sit at three extremes: (i) EHR representation learning methods, including sequential and graph-based models that encode temporal and heterogeneous clinical structures (Miotto et al., 2016; Li et al., 2020; Landi et al., 2020; Chen et al., 2024), (ii) text-to-SQL or graph semantic parsers (Wang et al., 2020; Lee et al., 2023; Raghavan et al., 2021; Bardhan et al., 2022), and (iii) retrieval-augmented pipelines built on large language models (LLMs) such as GPT-3.5/4 (Kweon et al., 2024; Elgedawy et al., 2024; Wu et al., 2024). These approaches can be accurate, yet they are computationally heavy, hard to deploy under strict privacy constraints, and largely ignore the strong structural priors present in EHR data.

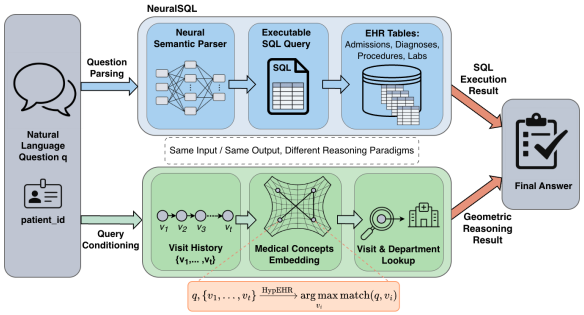

Figure 1: Comparison of workflows between text-to-SQL (*top*) and our method (*bottom*). The text-to-SQL-based methods typically rely on large-scale pretrained large language models to generate accurate SQL queries, whereas our method is specifically adapted to medical data, enabling comparable performance with a significantly smaller number of parameters.

Prior research in EHR representation learning indicates that medical codes and longitudinal patient trajectories are intrinsically hierarchical, exhibiting properties that align closely with hyperbolic geometry (Lu et al., 2019; Beaulieu-Jones et al., 2019). While Euclidean embeddings distort tree-like structures, hyperbolic spaces can embed hierarchies with arbitrarily low distortion. Building on this insight, Lu et al. (2023) demonstrate that hyperbolic embeddings of the medical code hierarchy can improve temporal health event prediction, but their resulting patient representations are ultimately modeled in a Euclidean space. This raises a central research question: can a compact model, explicitly aligned with the intrinsic geometry of EHRs at the patient level, compete with billion-parameter LLMs in complex question answering?

We address this question with **HypEHR** (**Hyp**erbolic modeling of **E**lectronic **H**ealth **R**ecords), a novel compact EHR-QA framework

*Accepted at Findings of ACL 2026

Findings of the Association for Computational Linguistics: ACL 2026, pages 10849–10862

based on hyperbolic clinical sequence modeling. The comparison of pipelines between HypEHR and text-to-SQL is shown in Figure 1. The resulting model achieves performance comparable to large language models, while being orders of magnitude smaller (22M) than typical LLM-based pipelines (trillions of parameters) and thus more suitable for on-premise, privacy-conscious clinical settings.

2 Methodology

2.1 Problem Definition and Model Overview

Given a question q and the visit history of patient $\mathcal{H}_p = \{v_1, \ldots, v_T\}$, where each v_t is associated with a set of medical concepts (including diagnosis codes, procedure codes, drug codes, admission time and other laboratory records), the model must look up the right visit, find the correct department, and return a clear answer. From the perspective of answer types, the common QA pairs can be categorized into four classes: boolean, concept, numerical, and integer count.

We summarize the main modules of the HypEHR framework in Figure 2 to provide an overview and the notations used throughout this paper in Table 3. HypEHR consists of two stages. The first stage, *patient encoder pretraining*, learns a hyperbolic patient encoder by joint targets of next-visit diagnosis prediction and hierarchy-aware regularization. The next stage, *question-answer training*, trains answer type-specified heads using embeddings from the frozen patient encoder.

2.2 Hyperbolic Clinical Sequence Encoder

We now present the first stage of our model — *patient encoder pretraining*. Each medical concept $c \in v_t$ is embedded into a Lorentzian hyperbolic manifold $\mathbb{H}^d_L$ via a learnable embedding function $e_c \in \mathbb{H}^d_L$. Within a visit, we aggregate the code embeddings using a hyperbolic attention mechanism to obtain a visit representation $h_t \in \mathbb{H}^d_L$. The sequence $\{h_t\}_{t=1}^T$ is then processed by a multi-layer Lorentz Transformer encoder, which adapts self-attention, residual connections, and normalization to the Lorentz manifold, yielding contextualized visit states $\{z_t\}_{t=1}^T$ and a global summary representation $z_{\text{[CLS]}} \in \mathbb{H}^d_L$. This global summary is then mapped to all diagnosis codes, producing next-visit diagnosis prediction loss $\mathcal{L}_{\text{diag}}$.

To encode hierarchical relationships in diagnosis code embeddings, we train patient encoder in a multi-task fashion with a hierarchy-aware regularizer $\mathcal{L}_{\text{hier}}$ using ICD code trie built by chapters → blocks → categories → subcategories, encouraging embeddings of codes that share ancestors in the ontology to be closer in hyperbolic distance than unrelated codes. We further decompose $\mathcal{L}_{\text{hier}}$ into a radial hierarchy term and a relative hierarchy term to jointly enforce depth ordering and local separation. Overall, the encoder parameters are optimized to minimize a joint objective

$$\mathcal{L} = \mathcal{L}_{\text{diag}} + \lambda \mathcal{L}_{\text{hier}} \tag{1}$$

where $\mathcal{L}_{\text{diag}}$ is the binary cross-entropy loss and

$$\mathcal{L}_{\text{hier}} = \mathcal{L}_{\text{rad}} + \mu\, \mathcal{L}_{\text{rel}}, \tag{2}$$

$$\mathcal{L}_{\text{rad}} = \sum_{(p,c)\in\mathcal{P}} \max\Big(0,\ \|e_p\|_{\mathbb{H}} - \|e_c\|_{\mathbb{H}} + \beta\Big), \tag{3}$$

$$\mathcal{L}_{\text{rel}} = \sum_{(a,a^+,a^-)\in\mathcal{T}} \max\Big(0,\ d_{\mathbb{H}}(e_a, e_{a^+}) - d_{\mathbb{H}}(e_a, e_{a^-}) + \alpha\Big) \tag{4}$$

with $\|e\|_{\mathbb{H}} := d_{\mathbb{H}}(e, \mathbf{o})$, and $\mathbf{o}$ denotes the origin of the Lorentz hyperboloid. Here $\mathcal{P}$ denotes parent–child pairs extracted from the ICD trie, and $\mathcal{T}$ denotes triplets where a^+ is an ancestor-related (e.g., parent/same-branch) code of a and a^- is a non-ancestor code; $\alpha, \beta > 0$ are margin hyperparameters, and $\lambda, \mu > 0$ balance the relative importance of each loss term. This yields a geometry-aware representation of patient trajectories that is later reused for downstream question answering.

2.3 Hyperbolic EHR-QA Model

Given a natural-language question q about a patient p, our model combines a natural language encoder with the Lorentzian patient encoder described above. The question q is first encoded by a biomedical pre-trained language encoder into a Euclidean vector $u_q \in \mathbb{R}^{d_e}$, which is then projected into the hyperbolic manifold via an affine map followed by an exponential map at the origin, yielding a question representation $z_q \in \mathbb{H}^d_L$. We then perform hyperbolic cross-attention from z_q over the sequence of visit states $\{z_t\}_{t=1}^T$: attention scores are defined as negative scaled hyperbolic distances $s_t = -\gamma d_{\mathbb{H}}(z_q, z_t)$ and normalized via softmax to obtain weights α_t. A hyperbolic weighted Fréchet mean of visit states, $z^{\text{visit}}_{p|q} = \text{HypAgg}(\{\alpha_t, z_t\}_{t=1}^T)$, serves as a question-conditioned patient summary. We implement Hyperbolic Aggregation (HypAgg)

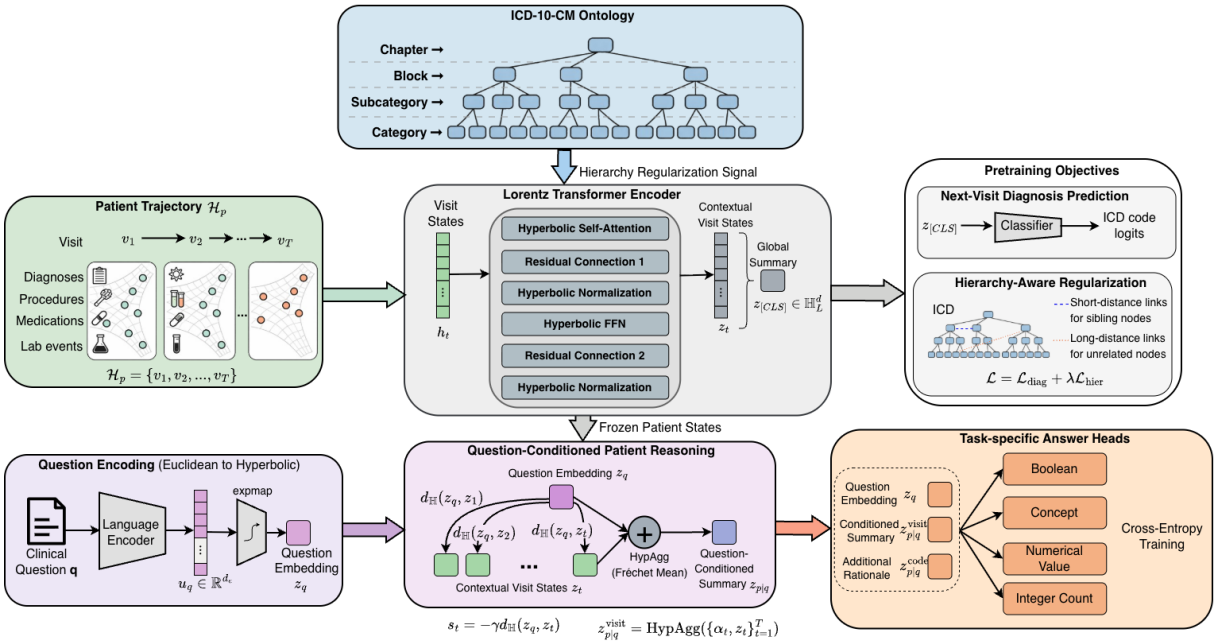

Figure 2: The overall framework of our proposed HypEHR.

as the exponential map at the origin of the weighted average of log-mapped points, which approximates the Riemannian barycenter on the Lorentz manifold. For fine-grained reasoning, we additionally apply a second-stage hyperbolic attention over code embeddings within the top-k attended visits to construct a code-level rationale vector $z_{p|q}^{\text{code}} \in \mathbb{H}_L^d$. Depending on the QA pair type, specialized answer heads consume $(z_q, z_{p|q}^{\text{visit}}, z_{p|q}^{\text{code}})$ to produce logits over the corresponding classes. The QA-specific components on top of the frozen language and patient encoders are trained using standard cross-entropy losses. Details of QA heads can be found in Appendix D.

3 Experiments

3.1 Experimental Setup

Datasets and Tasks We adopt two representative EHR QA datasets for training and evaluation: **MIMIC-IV-Ext-Instr** (Wu et al., 2024) and the tabular subset of **EHRXQA** (Bae et al., 2023), and report the accuracy(%) of the generated/retrieved answers. Besides, we also evaluate our model on four common clinical predictive tasks on **MIMIC-IV** (Johnson et al., 2023): (i) mortality prediction (MT), (ii) readmission prediction (RA), (iii) length-of-stay prediction (LOS), (iv) phenotype prediction (Pheno). AUPRC is adopted to evaluate the model's performance on the above classification tasks.

Baselines To comprehensively evaluate our proposed HypEHR, we adopt 6 representative methods as baselines for comparison from 3 main perspectives: (1) **text-to-SQL-based methods**: NeuralSQL (Bae et al., 2023) with GPT-5.2 (OpenAI, 2025) as the SQL parser, and a more lightweight version NeuralSQL-l with code-smol2-text-to-sql (Burtenshaw, 2024) as the SQL parser, (2) **LLM-based methods**: Llemr (Wu et al., 2024), EHRAgent (Shi et al., 2024) and Llama-3-8B (AI@Meta, 2024), where closed or open-source LLMs are used to process the question and structured patient history then generate answers, (3) **EHR representation learning-based methods**: a traditional patient sequence encoder RETAIN (Choi et al., 2016) is used as the patient encoder in our workflow. Results are the mean and standard deviation of 5 runs over different random seeds.

More details about data processing (including task definitions), baseline implementations, and hyperparameter tuning could be found in Appendix B.2, B.3, and E.

3.2 Experimental Results

Question-answering Results Table 1 presents the accuracy for each baseline. Since EHRXQA formulates questions as executable SQL queries, methods that explicitly leverage LLMs for SQL generation, such as NeuralSQL and EHRAgent, naturally align with this paradigm and therefore achieve superior performance. Notably, our method achieves the best performance among all approaches that do not rely on large language model–based frameworks (e.g., GPT-5.2), high-

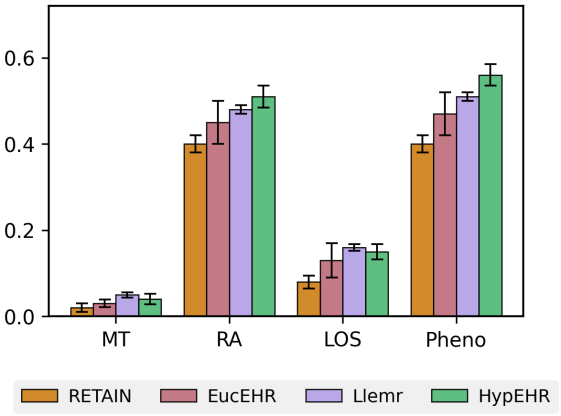

Figure 3: The AUPRC values of four models on the MIMIC-IV dataset.

lighting the potential of hyperbolic embeddings.

Model	EHRXQA	MIMIC-Instr
RETAIN	81.19 ±1.95	65.91 ±0.84
NeuralSQL	95.97 ±0.50	75.17 ±0.73
NeuralSQL-l	86.72 ±0.97	67.85 ±0.85
Llama-3	82.88 ±1.38	70.90 ±0.86
Llemr	87.25 ±0.77	77.53 ±0.54
EHRAgent	93.06 ±1.09	74.16 ±0.56
HypEHR	89.53 ±0.60	76.02 ±0.41

Table 1: Accuracy(%) of models across two QA datasets. **MIMIC-Instr** denotes MIMIC-IV-Ext-Instr. NeuralSQL (EHRXQA) and Llemr (MIMIC-Instr) are official baselines based on large-parameter LLMs, and therefore serve as approximate upper bounds for current performance on these datasets.

Clinical Prediction Results To further assess whether our hyperbolic patient encoder learns generally useful representations beyond EHR-QA, we attach simple classification heads for standard clinical prediction tasks and compare its performance against baselines. The results can be found in Figure 3. HypEHR achieves the best performance on readmission prediction and phenotype prediction, and demonstrates performance comparable to the LLM–based baseline Llemr, on remaining tasks.

Ablation Study To assess the contribution of each part attribute to model performance, we conduct an ablation study, evaluating HypEHR under different variants. Table 2 reports this study. The results highlight the relative importance of different components within the model architecture. Pretraining of the patient encoder plays the most critical role, and even under the same pretraining setting, the Euclidean model performs substantially worse than the hyperbolic model on the test set.

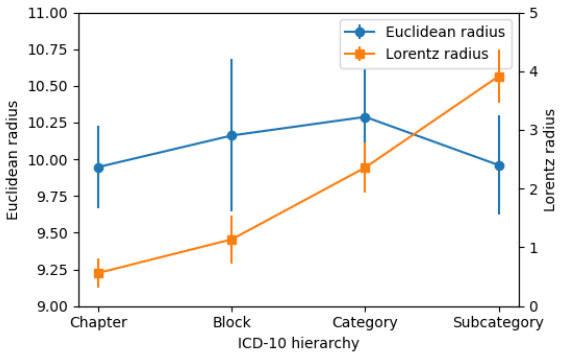

Figure 4: Comparison between Hyperbolic norms and Euclidean norms.

Although the hierarchy loss is not the primary contributor to overall model performance, it provides significant benefits in capturing and enforcing the hierarchical structure of codes.

Model	EHRXQA	MIMIC-Instr
w/o L_{hier}	82.72 ±3.41	70.38 ±0.54
w/o pretraining	74.05 ±4.76	68.12 ±1.39
EucEHR	80.33 ±1.14	69.88 ±1.07
HypEHR	89.53 ±0.60	76.02 ±0.41

Table 2: Ablation study on different variants of HypEHR. $\mathcal{L}_{\text{hier}}$ refers to Equation (2), pretraining denotes next-visit diagnosis prediction pretraining in Section 2.2, and EucEHR uses the same structure and pretraining task as HypEHR, but calculations are in Euclidean space.

Geometry Analysis To test whether embeddings reflect the intrinsic ICD diagnosis hierarchy, we sample codes beginning with `I`, group them by tree depth, and compute for each code c at depth k an embedding radius—$\|e_c^{\text{Euc}}\|_2$ for the Euclidean baseline and $r_c^{\text{Lor}} = d_{\mathbb{H}}(o, e_c^{\text{Lor}})$ for the Lorentz model—then average within each level to obtain $\bar{r}_k^{\text{Euc}}$ and $\bar{r}_k^{\text{Lor}}$. Figure 4 shows that Euclidean radii depend only weakly and noisily on depth (e.g., specific codes like `I21.9/I50.9` can have norms similar to `I10`), whereas the Lorentz model yields a clear monotonic increase, pushing deeper diagnoses farther from the origin; this radius–depth alignment suggests hyperbolic geometry better matches the tree-like expansion of the ICD hierarchy by allocating more capacity to fine-grained concepts near the boundary.

4 Conclusion

In this work, we revisited EHR question answering from the perspective of data geometry. We introduced HypEHR, a lightweight Lorentz-based

model that jointly encodes questions and clinical sequences. Experiments on MIMIC-IV-based QA benchmarks demonstrate comparative results with LLM baselines while using substantially fewer parameters.

Limitations

Our approach also has several limitations. First, it relies on a preprocessing step that restructures each dataset so that answers fall into a small set of predefined categories (e.g., boolean, categorical concept, integer count, numeric values), which introduces additional engineering effort and computational overhead. Second, the current framework is restricted to discriminative answer types and does not naturally handle more open-ended, generative responses, limiting its extensibility to free-form clinical question answering. Third, hyperbolic neural networks are computationally more complex than their Euclidean counterparts, and the relative immaturity of public hyperbolic geometry libraries can lead to implementation challenges and potential instability in large-scale training.

Potential Risks and Ethical Considerations

This work studies EHR question answering as a research problem and is not intended for direct clinical deployment. Potential risks include misinterpretation of model outputs if used for medical decision-making without proper clinical oversight, as well as biases inherited from retrospective EHR data. To mitigate these risks, our model is evaluated only on de-identified, publicly available datasets and does not provide diagnostic or treatment recommendations. We rely on the official de-identification procedures of these datasets and do not access or reconstruct any personally identifying information. We emphasize that such systems should be used as decision-support tools under human supervision rather than autonomous clinical agents. Future work should further investigate robustness, calibration, and fairness across patient subpopulations before real-world use.

Acknowledgments

This work was supported in part by the DOE SEACROGS project (DE-SC0023191) and the AFOSR project (FA9550-24-1-0231). Research reported in this work was also partially funded through a Patient-Centered Outcomes Research Institute (PCORI) Award 102727. The views in this work are solely the responsibility of the authors and do not necessarily represent the views of the Patient-Centered Outcomes Research Institute (PCORI), its Board of Governors or Methodology Committee.

References

AI@Meta. 2024. Llama 3 Model Card.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Seongsu Bae, Daeun Kyung, Jaehee Ryu, Eunbyeol Cho, Gyubok Lee, Sunjun Kweon, Jungwoo Oh, Lei Ji, Eric I.-Chao Chang, Tackeun Kim, and Edward Choi. 2023. EHRXQA: A Multi-Modal Question Answering Dataset for Electronic Health Records with Chest X-ray Images. *arXiv preprint*. ArXiv:2310.18652 [cs].

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 [cs].

Jayetri Bardhan, Kirk Roberts, and Daisy Zhe Wang. 2023. Question Answering for Electronic Health Records: A Scoping Review of datasets and models. *arXiv preprint*. ArXiv:2310.08759 [cs].

Brett K. Beaulieu-Jones, Isaac S. Kohane, and Andrew L. Beam. 2019. Learning Contextual Hierarchical Structure of Medical Concepts with Poincairé Embeddings to Clarify Phenotypes. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 24:8–17.

Martin R. Bridson and André Haefliger. 1999. Basic Concepts. In Martin R. Bridson and André Haefliger, editors, *Metric Spaces of Non-Positive Curvature*, pages 2–14. Springer, Berlin, Heidelberg.

Burtenshaw. 2024. burtenshaw/code-smol2-text-to-sql · Hugging Face.

Jiayuan Chen, Changchang Yin, Yuanlong Wang, and Ping Zhang. 2024. Predictive Modeling with Temporal Graphical Representation on Electronic Health Records. *arXiv preprint*. ArXiv:2405.03943 [cs].

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Ran Elgedawy, Ioana Danciu, Maria Mahbub, and Sudarshan Srinivasan. 2024. Dynamic Q&A of Clinical Documents with Large Language Models. *arXiv preprint*. ArXiv:2401.10733 [cs] version: 2.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. *arXiv preprint*. ArXiv:1804.01882 [cs].

M. Gromov. 1987. Hyperbolic Groups. In S. M. Gersten, editor, *Essays in Group Theory*, pages 75–263. Springer, New York, NY.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2018. Hyperbolic Attention Networks. *arXiv preprint*. ArXiv:1805.09786 [cs].

Neil He, Rishabh Anand, Hiren Madhu, Ali Maatouk, Smita Krishnaswamy, Leandros Tassiulas, Menglin Yang, and Rex Ying. 2025a. HELM: Hyperbolic Large Language Models via Mixture-of-Curvature Experts. *arXiv preprint*. ArXiv:2505.24722 [cs].

Neil He, Hiren Madhu, Ngoc Bui, Menglin Yang, and Rex Ying. 2025b. Hyperbolic Deep Learning for Foundation Models: A Survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 6021–6031. ArXiv:2507.17787 [cs].

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Liwei H. Lehman, Leo A. Celi, and Roger G. Mark. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1.

Sunjun Kweon, Jiyoun Kim, Heeyoung Kwak, Dongchul Cha, Hangyul Yoon, Kwanghyun Kim, Jeewon Yang, Seunghyun Won, and Edward Choi. 2024. EHRNoteQA: An LLM Benchmark for Real-World Clinical Practice Using Discharge Summaries. *arXiv preprint*. ArXiv:2402.16040 [cs].

Isotta Landi, Benjamin S. Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T. Dudley, Cesare Furlanello, and Riccardo Miotto. 2020. Deep representation learning of electronic health records to unlock patient stratification at scale. *npj Digital Medicine*, 3(1):96.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. 2023. EHRSQL: A Practical Text-to-SQL Benchmark for Electronic Health Records. *arXiv preprint*. ArXiv:2301.07695 [cs].

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

Yue Li, Haoxuan Qu, Mengyuan Liu, Jun Liu, and Yujun Cai. 2025. HyLiFormer: Hyperbolic Linear Attention for Skeleton-based Human Action Recognition. *arXiv preprint*. ArXiv:2502.05869 [cs].

Sichu Liang, Linhai Zhang, Hongyu Zhu, Wenwen Wang, Yulan He, and Deyu Zhou. 2025. RGAR: Recurrence Generation-augmented Retrieval for Factual-aware Medical Question Answering. *arXiv preprint*. ArXiv:2502.13361 [cs].

Chang Lu, Chandan K. Reddy, and Yue Ning. 2023. Self-Supervised Graph Learning with Hyperbolic Embedding for Temporal Health Event Prediction. *IEEE Transactions on Cybernetics*, 53(4):2124–2136. ArXiv:2106.04751 [cs].

Qiuhao Lu, Nisansa de Silva, Sabin Kafle, Jiazhen Cao, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Brent Hailpern, Berthold Reinwald, and Yunyao Li. 2019. Learning Electronic Health Records through Hyperbolic Embedding of Medical Ontologies. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '19, pages 338–346, New York, NY, USA. Association for Computing Machinery.

Riccardo Miotto, Li Li, Brian A. Kidd, and Joel T. Dudley. 2016. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports*, 6(1):26094.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. *arXiv preprint*. ArXiv:1705.08039 [cs].

OpenAI. 2024. Introducing ChatGPT.

OpenAI. 2025. Introducing GPT-5.2.

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 [cs].

Sarang Patil, Ashish Parmanand Pandey, Ioannis Koutis, and Mengjia Xu. 2025a. Hierarchical Mamba Meets Hyperbolic Geometry: A New Paradigm for Structured Language Embeddings. *arXiv preprint*. ArXiv:2505.18973 [cs].

Sarang Patil, Zeyong Zhang, Yiran Huang, Tengfei Ma, and Mengjia Xu. 2025b. Hyperbolic Large Language Models. *arXiv preprint*. ArXiv:2509.05757 [cs] version: 1.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. 2022. Hyperbolic Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10023–10044.

Preethi Raghavan, Jennifer J Liang, Diwakar Mahajan, Rachita Chandra, and Peter Szolovits. 2021. emrKBQA: A Clinical Knowledge-Base Question Answering Dataset. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 64–73, Online. Association for Computational Linguistics.

Erzsebet Ravasz and Albert-Laszlo Barabasi. 2002. Hierarchical Organization in Complex Networks.

Rik Sarkar. 2012. Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. In Marc Van Kreveld and Bettina Speckmann, editors, *Graph Drawing*, volume 7034, pages 355–366. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C. Ho, Carl Yang, and May Dongmei Wang. 2024. EHRAgent: Code Empowers Large Language Models for Few-shot Complex Tabular Reasoning on Electronic Health Records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22315–22339, Miami, Florida, USA. Association for Computational Linguistics.

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*, WWW '20, pages 350–361, New York, NY, USA. Association for Computing Machinery.

Zhenbang Wu, Anant Dadu, Mike Nalls, Faraz Faghri, and Jimeng Sun. 2024. Instruction Tuning Large Language Models to Understand Electronic Health Records.

Menglin Yang, Ram Samarth B. B, Aosong Feng, Bo Xiong, Jihong Liu, Irwin King, and Rex Ying. 2024. Hyperbolic Fine-Tuning for Large Language Models.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. 2025. Hypformer: Exploring Efficient Transformer Fully in Hyperbolic Space. *arXiv preprint*. ArXiv:2407.01290 [cs].

Qianru Zhang, Honggang Wen, Wei Yuan, Crystal Chen, Menglin Yang, Siu-Ming Yiu, and Hongzhi Yin. 2025. HMamba: Hyperbolic Mamba for Sequential Recommendation. *arXiv preprint*. ArXiv:2505.09205 [cs].

Zhengyun Zhao, Huaiyuan Ying, Yue Zhong, and Sheng Yu. 2025a. DR.EHR: Dense Retrieval for Electronic Health Record with Knowledge Injection and Synthetic Data. *arXiv preprint*. ArXiv:2507.18583 [cs].

Zhengyun Zhao, Hongyi Yuan, Jingjing Liu, Haichao Chen, Huaiyuan Ying, Songchi Zhou, Yue Zhong, and Sheng Yu. 2025b. CliniQ: A Multi-faceted Benchmark for Electronic Health Record Retrieval with Semantic Match Assessment. *arXiv preprint*. ArXiv:2502.06252 [cs].

A Notations

This section summarizes the notation used throughout the paper for clarity and ease of reference. Table 3 lists the definitions of all symbols appearing in the main text and appendices.

B Experiment Details

B.1 Implementation Details

All experiments were conducted using 4 NVIDIA A100 GPUs with 80GB memory each. Models were trained with a batch size of 48 using the geoopt.optim.RiemannianAdam[1] optimizer, and weight decay 1×10^{-2}. The learning rate was set to 3×10^{-4} with linear warmup over the first 10% of training steps, followed by cosine decay. Gradient norms were clipped to a maximum ℓ_2 norm of 1.0. Dropout with rate 0.1 was applied to attention weights, feed-forward layers, and residual connections. Training used early stopping based on validation loss with a patience of 10 epochs for both patient encoder pretraining and question-answering head training. The maximum number of epochs was set to 250 for pretraining (approximately 2 hours) and 200 for question-answering heads (approximately 5 minutes), and the checkpoint with the best validation performance was selected for downstream evaluation.

The Lorentz Transformer encoder consisted of 3 layers, each with 6 attention heads. The hyperbolic embedding dimension was set to 390, consistent with He et al. (2025a). The total number of trainable parameters in the full model was approximately 22 million, corresponding to a model size of 84 MB when stored in 32-bit floating point format. All models were implemented in PyTorch[2] and leveraged Geoopt[3] for Riemannian optimization on the Lorentz manifold. Mixed-precision training (FP16) was enabled via NVIDIA Apex to reduce memory usage and improve throughput. For text encoder, we use Bio_ClinicalBERT (Alsentzer et al., 2019)[4].

B.2 Data Preprocessing

EHRXQA EHRXQA is a multi-modal EHR question answering dataset that links MIMIC-IV structured tables with aligned MIMIC-CXR chest

[1] https://geoopt.readthedocs.io/en/latest/optimizers.html
[2] https://pytorch.org
[3] https://github.com/geoopt/geoopt
[4] https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT.

Symbol	Meaning
p	Patient index.
q	Natural-language question about patient p.
$\mathcal{H}_p = \{v_1, \ldots, v_T\}$	Visit history (trajectory) of patient p.
T	Number of visits in the trajectory.
v_t	The t-th visit.
$c \in v_t$	A medical concept/code occurring in visit v_t.
d	Hyperbolic embedding dimension (Lorentz model).
d_e	Euclidean embedding dimension of the text encoder output.
$\mathbb{H}^d_L$	d-dimensional hyperbolic space in the Lorentz (hyperboloid) model.
$\mathbb{R}^d$	Euclidean Tangent Space $\mathbb{H}^d_L \cong \mathbb{R}^d$.
o	Origin point on the Lorentz hyperboloid (used for exp/log maps and radii).
$d_{\mathbb{H}}(\cdot,\cdot)$	Hyperbolic geodesic distance on $\mathbb{H}^d_L$.
$e_c \in \mathbb{H}^d_L$	Hyperbolic embedding of medical concept/code c.
$\mathrm{h_t} \in \mathbb{H}^d_L$	Visit representation aggregated from code embeddings within v_t.
$\{z_t\}_{t=1}^T \subset \mathbb{H}^d_L$	Contextualized visit states from the Lorentz Transformer encoder.
$z_{[\mathrm{CLS}]} \in \mathbb{H}^d_L$	Global patient summary representation (CLS token/state).
$u_q \in \mathbb{R}^{d_e}$	Euclidean question representation from the language encoder (pooled).
$u_i \in \mathbb{R}^{d_e}$	Euclidean token embedding of the i-th question token.
$\tilde{u}_i = Wu_i + b$	Affine projection from Euclidean text space to tangent space at o.
$\exp_o(\cdot), \log_o(\cdot)$	Exponential and logarithmic maps between $\mathbb{R}^d$ and $\mathbb{H}^d_L$.
$z^q_i = \exp_o(\tilde{u}_i) \in \mathbb{H}^d_L$	Hyperbolic embedding of question token i.
$z_q \in \mathbb{H}^d_L$	Pooled hyperbolic question embedding (e.g., via Hyperbolic aggregation).
γ	Cross-attention temperature/scale in scores $s_t = -\gamma d_{\mathbb{H}}(z_q, z_t)$.
α_t	Attention weight over visits (softmax-normalized).
$\mathrm{HypAgg}(\cdot)$	Hyperbolic aggregation operator (approx. Fréchet mean on $\mathbb{H}^d_L$).
$z^{\mathrm{visit}}_{p\|q} \in \mathbb{H}^d_L$	Question-conditioned visit-level patient summary.
$z^{\mathrm{code}}_{p\|q} \in \mathbb{H}^d_L$	Question-conditioned code-level patient summary / rationale vector.
L_{diag}	Next-visit diagnosis prediction loss (binary cross-entropy).
L_{hier}	Hierarchy-aware regularization loss for ICD code embeddings.
L_{rad}	Radial hierarchy term encouraging depth ordering by hyperbolic radius.
L_{rel}	Relative hierarchy term enforcing ancestor vs non-ancestor separation.
λ	Weight on hierarchy regularization in $L = L_{\mathrm{diag}} + \lambda L_{\mathrm{hier}}$.
μ	Weight on relative term in $L_{\mathrm{hier}} = L_{\mathrm{rad}} + \mu L_{\mathrm{rel}}$.
β	Margin in radial hierarchy loss L_{rad}.
α	Margin in relative hierarchy loss L_{rel}.
$\mathcal{P}_{\mathrm{ICD}}$	Set of (parent, child) pairs extracted from the ICD trie.
$\mathcal{T}_{\mathrm{ICD}}$	Set of triplets (a, a^+, a^-) for relative hierarchy training.
$\|e\|_{\mathbb{H}} := d_{\mathbb{H}}(e, o)$	Hyperbolic radius (distance to origin).
$C_p = \{c_1, \ldots, c_K\}$	Per-patient candidate concept set for concept QA (plus optional null).
c_{null}	Learned "no-answer" pseudo-concept for concept QA.
$E = \{e_1, \ldots, e_M\}$	Candidate numeric events for a variable (e.g., lab test events).
e_j	Numeric event with timestamp t_j, value ν_j, and embedding h^{val}_j.
e_{null}	Learned null event for no-answer numeric questions.
e^{Euc}_c	Euclidean embedding of code c (for the EucEHR baseline).
e^{Lor}_c	Lorentz embedding of code c (for HypEHR).
r^{Lor}_c	Hyperbolic radius of code c, defined as $d_{\mathbb{H}}(o, e^{\mathrm{Lor}}_c)$.
$\bar{r}^{\mathrm{Euc}}_k$	Average Euclidean norm of codes at tree depth k.
$\bar{r}^{\mathrm{Lor}}_k$	Average hyperbolic radius of codes at tree depth k.
ϵ	Tolerance for matching a gold numeric value to candidate events.
$\mathcal{I} = \{j : \|\nu_j - \nu\| < \epsilon\}$	Index set of events matching target value ν.
$K_{\max}$	Maximum discretized count for the count head.

Table 3: Notation used throughout the paper. We use the Lorentz (hyperboloid) model $\mathbb{H}^d_L$ for hyperbolic embeddings and denote the hyperbolic distance by $d_{\mathbb{H}}$.

X-ray images to generate Image-, Table-, and Image+Table QA pairs requiring both unimodal and cross-modal reasoning. In our experiments, we use the tabular subset of EHRXQA, and categorize these questions to Boolean Value, Count, Float Value, and Concept according to their answer type. The train/valid/test split is provided in json files in the dataset.

MIMIC-IV-Ext-Instr We use the Schema Alignment subset of MIMIC-IV-Ext-Instr and adopt the same preprocessing procedure as EHRXQA. In addition, our training, validation, and test splits follow those used in Llemr (Wu et al., 2024).

MIMIC-IV We follow the data preprocessing process of Chen et al. (2024), filtering out patients with less than two visits. ICD-9-CM codes are mapped to unique ICD-10-CM codes by General Equivalence Mappings (GEMs)[5]. Statistics of processed data is shown in Table 4.

Dataset	MIMIC-IV
# of patients	14,155
# of visits	42,053
Avg. # of visits per patient	2.97
Max # of visits per patient	70
# of unique diagnoses	11,225
# of unique procedures	8,352
# of unique medicines	196

Table 4: Statistics of MIMIC-IV after pre-processing.

B.3 Baseline Implementations

- RETAIN (Choi et al., 2016) is a classical model for patient modeling. We use RETAIN to replace the patient encoder in our model, serving as a traditional baseline model.
- NeuralSQL is the standard baseline in EHRXQA (Bae et al., 2023), using gpt-3.5-turbo-0613 (OpenAI, 2024) to generate SQL queries then retrieve answer from database. We replaced gpt-3.5-turbo-0613 by SOTA model GPT-5.2 (OpenAI, 2025). NeuralSQL-*l* is a light-weighted version where a text-to-SQL specified small language model codesmol2-text-to-sql (Burtenshaw, 2024) serves as the text-to-SQL parser.
- EHRAgent (Shi et al., 2024) is an EHR question-answering intelligent agent equipped with a Python code interface and tool calling capabilities.
- Llama-3 (AI@Meta, 2024) is a compact, open-weight large language model. We directly use the problem and related patient history codes as a prompt to generate the answer.
- Llemr (Wu et al., 2024): Llemr is an instruction-tuned large language model framework that enables LLMs to process and interpret complex EHR data for diverse clinical question answering and predictive tasks. We use the pre-trained weights provided by the authors[6].

C Additional Results

Accuracy of Each Type of Question For all question types (Boolean, single-concept, numerical, and count queries, including no-answer cases), we report exact match accuracy, i.e., the proportion of questions for which the predicted answer array exactly matches the gold answer array. Results are shown in Table 5 and 6.

D Model Details

D.1 Answer Types and Prediction Heads

In this work, we focus on *per-patient* EHR-QA and restrict ourselves to the following answer categories:

- **Boolean / existence questions:** answers of the form [0] or [1], denoting *false* or *true*. These questions ask whether a given condition, procedure, or event exists in the patient record.
- **Concept questions:** answers as a single-element string array, e.g., ["pneumonia"] or ["low lung volumes"]. We align such strings to a discrete concept vocabulary (e.g., ICD, LOINC, or a curated set of findings) whenever possible.
- **Numeric value questions:** answers as a single floating-point value array, e.g., [5.0], corresponding to a laboratory test result or scalar measurement.

[5]The ICD-9-CM to ICD-10-CM General Equivalence Mappings (GEMs) are provided by the Centers for Medicare & Medicaid Services (CMS) and made available via the National Bureau of Economic Research (NBER): https://www.nber.org/research/data/icd-9-cm-and-icd-10-cm-and-icd-10-pcs-crosswalk-or-general-equivalence-mappings.

[6]https://github.com/zzachw/llemr.

Model	BL	CT	FL	CP	Overall
RETAIN	84.36 ± 3.46	82.15 ± 5.29	78.22 ± 4.23	80.74 ± 1.20	81.19 ± 1.95
NeuralSQL	96.44 ± 1.42	95.71 ± 0.96	94.97 ± 0.58	96.87 ± 1.07	95.97 ± 0.50
NeuralSQL-l	88.40 ± 2.37	87.12 ± 1.98	84.53 ± 2.25	87.23 ± 0.97	86.72 ± 0.97
Llama-3	86.40 ± 2.25	81.56 ± 3.47	81.09 ± 2.88	83.22 ± 1.98	82.88 ± 1.38
Llemr	87.85 ± 1.54	88.96 ± 2.01	84.23 ± 0.94	88.21 ± 1.46	87.25 ± 0.77
EHRAgent	92.46 ± 1.14	94.50 ± 2.98	93.26 ± 1.84	91.89 ± 2.08	93.06 ± 1.09
HypEHR	91.54 ± 1.74	89.41 ± 0.97	90.48 ± 0.78	87.05 ± 1.33	89.53 ± 0.60

Table 5: Accuracy(%) for each question type on EHRXQA. Reported values are mean ± std. **BL**: Boolean Value, **CT**: Count, **FL**: Float Value, **CP**: Concept. Weights: (BL, CT, FL, CP) = (0.21, 0.26, 0.27, 0.26).

Model	BL	CT	FL	CP	Overall
RETAIN	66.37 ± 1.85	64.29 ± 2.00	65.62 ± 1.06	67.33 ± 1.94	65.91 ± 0.84
NeuralSQL	74.32 ± 0.87	76.03 ± 1.14	75.11 ± 1.94	75.33 ± 1.23	75.17 ± 0.73
NeuralSQL-l	69.64 ± 1.56	68.47 ± 1.75	67.86 ± 1.64	65.29 ± 1.84	67.85 ± 0.85
Llama-3	70.35 ± 1.74	71.54 ± 2.01	69.98 ± 1.47	72.09 ± 1.75	70.90 ± 0.86
EHRAgent	74.35 ± 1.25	73.64 ± 1.42	72.54 ± 0.89	76.55 ± 0.97	74.16 ± 0.56
Llemr	77.58 ± 0.95	76.21 ± 1.04	77.85 ± 1.21	78.34 ± 0.96	77.53 ± 0.54
HypEHR	76.51 ± 0.75	77.32 ± 0.98	74.46 ± 0.90	76.28 ± 0.43	76.02 ± 0.41

Table 6: Accuracy(%) for each question type on MIMIC-IV-Ext-Instr. Reported values are mean ± std. **BL**: Boolean Value, **CT**: Count, **FL**: Float Value, **CP**: Concept. Weights: (BL, CT, FL, CP) = (0.25, 0.22, 0.30, 0.23).

- **Count questions:** answers as a single integer array $[k]$ with $k > 1$, e.g., the count of events or measurements matching a condition.
- **No-answer cases:** answers as an empty array $[]$, indicating that *no* event in the patient record satisfies the query.

We explicitly exclude questions whose answers are *lists of patient identifiers*, such as $[10501557, 12215941, \dots]$, since these correspond to cohort-level retrieval rather than single-patient QA and are outside our scope.

Question-conditioned hyperbolic representations. All heads operate on a shared set of hyperbolic representations. Let $\{z_t\}_{t=1}^{T} \subset \mathbb{H}_L^d$ denote the Lorentzian visit-level embeddings of a patient's trajectory, and $z_{[\text{CLS}]} \in \mathbb{H}_L^d$ the global patient representation produced by our Lorentz Transformer encoder. A natural-language question q is encoded by a text encoder into Euclidean token embeddings $\{u_i\}$, which are mapped into the Lorentz model via the exponential map at the origin:

$$\tilde{u}_i = W u_i + b, \quad z_i^q = \exp_o(\tilde{u}_i) \in \mathbb{H}_L^d. \tag{5}$$

We obtain a pooled question representation $z_q \in \mathbb{H}_L^d$ via hyperbolic aggregation of $\{z_i^q\}$.

We then compute a question-conditioned patient summary at the visit level using hyperbolic attention:

$$\alpha_t \propto \exp\big(-\gamma\, d_{\mathbb{H}}(z_q, z_t)\big) \tag{6}$$

$$z_{p|q}^{\text{visit}} = \text{HypAgg}\big(\{\alpha_t, z_t\}_{t=1}^{T}\big) \tag{7}$$

where $d_{\mathbb{H}}(\cdot,\cdot)$ is the Lorentzian distance and HypAgg is a Fréchet mean operator on $\mathbb{H}_L^d$. For concept-level reasoning, we further refine attention to the code-level within the top-k attended visits, yielding a code-level summary $z_{p|q}^{\text{code}} \in \mathbb{H}_L^d$.

For all prediction heads, we map hyperbolic vectors back to the tangent space at the origin via the logarithmic map,

$$\hat{z} = \log_o(z) \in \mathbb{R}^d, \tag{8}$$

and feed $\hat{z}$ (optionally concatenated with other features) into Euclidean MLPs.

D.1.1 Boolean / Existence Head

This head handles questions whose answers are encoded as $[0]$ or $[1]$.

Input. We concatenate the question-conditioned patient representation and the question embedding in tangent space:

$$h_{\text{bool}} = \hat{z}_{p|q}^{\text{visit}} \oplus \hat{z}_q \in \mathbb{R}^{2d}. \tag{9}$$

Output. A small MLP produces logits $o \in \mathbb{R}^2$ for the labels "no" and "yes":

$$o = \mathrm{MLP}_{\text{bool}}(h_{\text{bool}}), \quad p = \mathrm{softmax}(o), \quad (10)$$

where p_y denotes the predicted probability of label $y \in \{0, 1\}$.

Loss. With ground-truth $y \in \{0, 1\}$ derived from $[0]/[1]$, the loss is standard cross-entropy:

$$\mathcal{L}_{\text{bool}} = -\log p_y. \quad (11)$$

If a boolean-type question is annotated with an empty array [], we normalize it to $y = 0$ during preprocessing.

D.1.2 Concept Head

This head is used for questions whose answer is a single concept string, e.g., a diagnosis or finding.

Candidate set. For each patient, we construct a per-patient candidate set

$$\mathcal{C}_p = \{c_1, \ldots, c_K\}, \quad (12)$$

containing all concepts (codes or findings) appearing in that patient's EHR, plus an optional learned "no-answer" pseudo-concept c_{null}. Each candidate concept c_j has a hyperbolic embedding $e_{c_j} \in \mathbb{H}^d_L$.

Input and scoring. We use the question-conditioned code-level representation $z^{\text{code}}_{p|q}$ and compute pairwise scores against each candidate:

$$\phi_j = \mathrm{MLP}_{\text{pair}}\big(\log_o(z^{\text{code}}_{p|q}) \oplus \log_o(e_{c_j})\big)$$

$$s_j = w^\top \phi_j \quad (13)$$

Output. We apply a softmax over all candidates:

$$p_j = \frac{\exp(s_j)}{\sum_k \exp(s_k)}. \quad (14)$$

At inference time, we select $\hat{c} = \arg\max_j p_j$ and output its associated string.

Loss. Let c be the target concept aligned from the answer string, and j its index in $\mathcal{C}_p$ (or the index of c_{null} if the answer is empty):

$$\mathcal{L}_{\text{concept-1}} = -\log p_j. \quad (15)$$

D.1.3 Float Value Head

This head handles questions whose answers are single numeric values, typically derived from laboratory tests or scalar measurements.

Event candidates. For a given variable (e.g., creatinine), we collect all matching events in the patient record:

$$\mathcal{E} = \{e_1, \ldots, e_M\}, \quad (16)$$

where each event e_j has a timestamp t_j, a scalar value ν_j, and a hyperbolic embedding $h^{\text{val}}_j \in \mathbb{H}^d_L$ (e.g., obtained from the corresponding visit state and variable identity). We additionally introduce a learned "null event" e_{null} for no-answer cases.

Input and scoring. We use the question-conditioned visit-level representation $z^{\text{visit}}_{p|q}$ and compute pairwise scores:

$$\phi_j = \mathrm{MLP}_{\text{val}}\big(\log_o(z^{\text{visit}}_{p|q}) \oplus \log_o(h^{\text{val}}_j)\big),$$

$$s_j = w^\top \phi_j \quad (17)$$

Output. We apply a softmax over all candidate events (including the null event):

$$p_j = \frac{\exp(s_j)}{\sum_k \exp(s_k)}. \quad (18)$$

The predicted answer is the value associated with the selected event, i.e., $\hat{v} = \nu_{\hat{j}}$ where $\hat{j} = \arg\max_j p_j$.

Loss. We treat numeric value prediction as a pointer-selection problem. Given a ground-truth value ν, we align it to one or more events in $\mathcal{E}$:

$$\mathcal{I} = \{j \mid |\nu_j - \nu| < \epsilon\}, \quad (19)$$

for a small tolerance ϵ. If at least one matching event exists, the loss is a multi-positive log-loss:

$$\mathcal{L}_{\text{value}} = -\log \sum_{j \in \mathcal{I}} p_j. \quad (20)$$

If no event in $\mathcal{E}$ matches the target value (or the gold answer is an empty array), we set $\mathcal{I} = \{j_{\text{null}}\}$ to select the null event.

D.1.4 Count Head

The count head is responsible for questions that ask for the number of events, visits, or occurrences satisfying a condition.

Input. We again use the question-conditioned visit-level representation:

$$h_{\text{count}} = \hat{z}^{\text{visit}}_{p|q} \oplus \hat{z}_q. \quad (21)$$

Output. We discretize counts into $\{0, 1, \ldots, K_{\max}\}$, where $K_{\max}$ is chosen based on the empirical distribution (e.g., a high percentile). The head outputs logits $o \in \mathbb{R}^{K_{\max}+1}$:

$$o = \mathrm{MLP}_{\text{count}}(h_{\text{count}}), \quad p = \mathrm{softmax}(o), \quad (22)$$

where p_k denotes the predicted probability of count k.

Loss. For a ground-truth count k (clipped to $K_{\max}$ if necessary), we use cross-entropy:

$$\mathcal{L}_{\text{count}} = -\log p_k. \quad (23)$$

When the original answer is an empty array for a count-type question, we normalize it to $k = 0$ during preprocessing.

D.1.5 Overall Objective

For each question, exactly one head is activated based on the parsed answer type. The total QA loss aggregates the head-specific losses together with auxiliary pretraining and geometry-aware regularization terms, where only the relevant terms are present for each sample. During QA training, only head-specific losses are optimized; pretraining losses are inactive due to encoder freezing.

E Hyperparameter Tuning

We tune hyperparameters on the validation split of each dataset. Our training has two stages. In Stage 1 we train the hyperbolic patient encoder with the objective $\mathcal{L} = \mathcal{L}_{\text{diag}} + \lambda \mathcal{L}_{\text{hier}}$, where $\mathcal{L}_{\text{hier}} = \mathcal{L}_{\text{rad}} + \mu \mathcal{L}_{\text{rel}}$. In Stage 2 we freeze the patient encoder and train the QA heads for each answer type. Best-found hyperparameter values are underlined in each setting.

Stage 1: Patient Encoder Pretraining We select hyperparameters by minimizing the validation value of $\mathcal{L}$. We tune: (i) the number of Lorentz Transformer encoder layers L; (ii) the hierarchy loss weight λ; (iii) the relative-term weight μ in $\mathcal{L}_{\text{hier}}$; (iv) the margin parameters α and β used in $\mathcal{L}_{\text{rel}}$ and $\mathcal{L}_{\text{rad}}$; (v) the learning rate for pretraining, and (vi) the hidden dimension d. We run a grid search over the following sets: $L \in \{\underline{3}, 5, 8\}$, $\lambda \in \{0, 0.1, \underline{0.5}, 1.0, 2.0\}$, $\mu \in \{0.25, \underline{0.5}, 1.0\}$, $\alpha \in \{0.1, \underline{0.2}, 0.5, 1.0\}$, $\beta \in \{0.1, \underline{0.2}, 0.5, 1.0\}$, $\eta \in \{10^{-5}, 3 \cdot 10^{-5}, \underline{10^{-4}}, 3 \cdot 10^{-4}\}$, $d \in \{130, 260, \underline{390}, 520\}$. After selection, we train the patient encoder once on the union of the training and validation sets for the same number of epochs, and we keep the final checkpoint for QA training.

Stage 2: Question Answering Heads Stage 2 uses the frozen patient encoder and a frozen text encoder, and trains only the QA-specific modules. We tune hyperparameters by maximizing validation exact-match accuracy on each QA dataset. We tune: (i) the cross-attention distance scale γ in $\alpha_t \propto \exp(-\gamma\, d_H(z_q, z_t))$; (ii) the number of top-k attended visits used for the optional code-level attention; and (iii) the learning rate for QA-head training. We search $\gamma \in \{0.5, \underline{1}, 2, 5\}$, $k \in \{1, 2, \underline{4}, 8\}$, and $\eta_{\text{QA}} \in \{10^{-5}, \underline{3 \cdot 10^{-5}}, 10^{-4}\}$. For all other settings (batch size, optimizer type, and training epochs) we keep the same values across runs to isolate the effect of the tuned parameters.

Random seeds For the final reported numbers, we rerun training with five random seeds $(42, 24, 33, 55, 67)$ and report the mean and standard deviation.

F Related Work

F.1 EHR Question Answering

Prior work on EHR question answering (EHR-QA) spans unstructured clinical notes, structured patient records, and multimodal combinations. For note-centric QA, emrQA (Pampari et al., 2018) constructs question-answer pairs from clinical annotations, while EHRNoteQA (Kweon et al., 2024) targets patient-specific, multi-note reasoning grounded in real clinical queries. Structured QA is studied in emrKBQA (Raghavan et al., 2021), which maps questions to executable logical forms over EHR knowledge bases, as well as text-to-SQL benchmarks such as EHRSQL (Lee et al., 2023). Multimodal datasets integrate notes and tables, including DrugEHRQA (Bardhan et al., 2022) and EHRXQA (Bae et al., 2023). In parallel, retrieval-focused benchmarks and models—such as CliniQ (Zhao et al., 2025b), DR.EHR (Zhao et al., 2025a), and retrieval-augmented methods like RGAR (Liang et al., 2025)—demonstrate that effective EHR-QA critically relies on accurate patient-specific evidence retrieval.

F.2 Hyperbolic Neural Networks

Transformer-style models in hyperbolic geometry largely build on the idea that hierarchical or power-law structure can be represented more naturally in negatively curved spaces (Ravasz and

Barabasi, 2002; Nickel and Kiela, 2017; Yang et al., 2024). Early work, such as Hyperbolic Attention Networks (Gulcehre et al., 2018), introduced hyperbolic variants of the attention mechanism and demonstrated how Transformer-like attention can be reformulated beyond Euclidean dot products. Subsequent efforts involved moving from hyperbolic attention inside an otherwise Euclidean Transformer toward more complete architectures. Hypformer (Yang et al., 2025) proposes a Transformer defined end-to-end in the Lorentz model and further develops linear-time hyperbolic self-attention for scalability, enabling billion-scale graph processing. More recent advances include HyLiFormer (Li et al., 2025), which introduces hyperbolic linear attention for efficient hierarchical sequence modeling, and HELM (He et al., 2025a), which trains fully hyperbolic LLMs using a mixture of curvature experts and hyperbolic multi-head latent attention to align geometric representations with semantic hierarchies. Parameter-efficient adaptation methods such as HypLoRA (Yang et al., 2024) enable hyperbolic fine-tuning of pre-trained models with up to 13% improvement on mathematical reasoning tasks. Beyond Transformer architectures, hyperbolic geometry has been integrated with state-space models for efficient sequence modeling: Hierarchical Mamba (HiM) (Patil et al., 2025a) combines Mamba's linear-time complexity with learnable hyperbolic curvature for hierarchical reasoning, while HMamba (Zhang et al., 2025) applies hyperbolic geometry to sequential recommendation with curvature-normalized discretization. Recent surveys have begun to categorize emerging hyperbolic deep learning architectures (Peng et al., 2022; He et al., 2025b; Patil et al., 2025b) across domains, tasks, and hyperbolic implementation approaches. However, despite these advances, the application of hyperbolic geometry to clinical question answering over structured EHR data remains largely unexplored.

G Geometry Hypothesis Validation

We justify our hypothesis about the EHR sequence using a Lorentzian hyperbolic space as the representation space for ICD-10-CM codes in our patient model by proposing the following theorem:

Proposition 1 (Hyperbolic suitability of the ICD-10-CM hierarchy)**.** *Let $\mathcal{C}$ be the set of ICD-10-CM diagnosis codes used in our study, and let d_T be the tree metric induced by the official parent–child hierarchy (each non-root code has a unique parent by truncating its code string to a more general prefix). Consider the d-dimensional Lorentzian hyperbolic space*

$$\mathbb{H}^d_L \;=\; \big\{x \in \mathbb{R}^{d+1} : \langle x, x\rangle_L = -1,\; x_0 > 0\big\},$$

where $\langle\cdot,\cdot\rangle_L$ is the Minkowski bilinear form, equipped with the induced hyperbolic distance $d_{\mathbb{H}}$. Then:

1. *The metric space $(\mathcal{C}, d_T)$ is 0-hyperbolic in the sense of Gromov (i.e., a metric tree).*
2. *For any $\varepsilon > 0$ there exists a dimension $d \geq 2$ and an embedding $\varphi : \mathcal{C} \to \mathbb{H}^d_L$ such that for all $u, v \in \mathcal{C}$,*

$$\begin{aligned}(1-\varepsilon)\, d_T(u,v) &\leq d_{\mathbb{H}}\big(\varphi(u), \varphi(v)\big) \\ &\leq (1+\varepsilon)\, d_T(u,v)\end{aligned} \tag{24}$$

In particular, the ICD-10-CM hierarchy admits a low-distortion embedding into the Lorentz model of hyperbolic space, so hyperbolic distances between code embeddings can faithfully reflect their hierarchical separation.

The geometric justification for this theorem lies in the exponential growth of volume in hyperbolic space, which naturally accommodates the exponential expansion of nodes in a hierarchy (chapters → blocks → categories).

Proof. (1) The ICD-10-CM tabular list organizes diagnosis codes into a rooted hierarchy (chapters → blocks → categories → subcategories), with each non-root code having a unique parent obtained by truncating its prefix. Taking $\mathcal{C}$ as vertices and connecting each code to its unique parent yields a connected, acyclic, rooted graph, hence a simplicial tree T. Endowing T with the path metric d_T makes $(\mathcal{C}, d_T)$ a geodesic metric tree. By the standard characterization of geodesic metric trees as precisely the 0-hyperbolic geodesic spaces (Gromov, 1987; Bridson and Haefliger, 1999), $(\mathcal{C}, d_T)$ is 0-hyperbolic.

(2) Results on embeddings of tree metrics into hyperbolic space show that any finite tree $(\mathcal{C}, d_T)$ admits, for every $\varepsilon > 0$, a $(1+\varepsilon)$-bilipschitz embedding into the hyperbolic plane $\mathbb{H}^2$; see, for example, Sarkar's construction of low-distortion Delaunay embeddings of trees in the hyperbolic plane (Sarkar, 2012). Concretely, there exists $\psi : \mathcal{C} \to \mathbb{H}^2$ such that for all $u, v \in \mathcal{C}$, quasi-isometry (24) holds.

The Lorentz hyperboloid model $\mathbb{H}^d_L$ is isometric to other standard models of hyperbolic space (such as the Poincaré ball and half-space models) via smooth bijections that preserve geodesic distance (Bridson and Haefliger, 1999; Nickel and Kiela, 2017; Ganea et al., 2018). Extending ψ to dimension $d \geq 2$ and composing with such an isometry yields an embedding $\varphi : \mathcal{C} \to \mathbb{H}^d_L$ satisfying the same bilipschitz bounds. This establishes item (2) and completes the proof. $\square$