EVERYQUERY: A PROMPTABLE FOUNDATION MODEL FOR CLINICAL PREDICTION TASKS OVER EHRS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive foundation models for electronic health records (EHRs) can make predictions for diverse clinical tasks without finetuning (Waxler et al., 2025; Renc et al., 2024; 2025). These methods work by generating possible "synthetic futures" for patients, and then inferring downstream predictions over these simulated trajectories. Despite their strong capabilities, these methods have several flaws. (1) Inference is extremely computationally expensive, as each prediction requires simulating many trajectories, each with many observations. (2) Their performance is noisy due to the variance inherent to simulation, which in particular reduces efficacy on rare events, which are often of high importance clinically. (3) They are not promptable, as their only input is the patient's medical history, requiring indirect aggregation at inference time. We introduce EveryQuery, a promptable foundation model for structured EHR data. Building on the insight that clinical prediction tasks can be written in a structured query language, EveryQuery takes a query as a prompt alongside the patient's history and estimates the outcome directly. This allows EveryQuery to make predictions across diverse clinical tasks, such as classification and survival modeling, without simulation or finetuning. Across three EHR datasets, EveryQuery outperforms a competitive autoregressive baseline on clinically relevant classification and time-to-event tasks, with mean AUROC gains of 14.3% and 12.7% on MIMIC-IV and a large academic medical center and comparable performance on the smaller NWICU dataset. Its advantage is largest for rare outcomes, and it requires $\sim 580\times$ less inference compute.

1 INTRODUCTION

Foundation models for structured electronic health record (EHR) data are an emerging new paradigm in clinical artificial intelligence (AI). Much like in other AI domains, foundation models for EHR data are appealing because they can make predictions across a diverse range of clinically meaningful tasks, such as classification or survival analysis. Currently, the only models capable of making predictions for diverse tasks at inference without finetuning are autoregressive (AR), generative models. Across numerous datasets, AR models show strong performance in estimating diverse clinical outcomes (Waxler et al., 2025; Renc et al., 2024; 2025; Rajamohan et al., 2025; Shmatko et al., 2025; McDermott, 2025).

Similar to large language models, autoregressive models are trained with a next-token prediction objective. At each training step, they perform a supervised multiclass classification problem over the model's vocabulary to predict the next medical event. Operationally, they split a patient's record into a past and future, and use the patient's past to predict which code comes first in their future. Unlike LLMs, however, which can answer novel questions directly at inference time through task prompting, AR models make predictions for specific tasks through simulation-based inference. Given as input a patient's medical history, they generate possible "synthetic futures" for the patient. The user must then extract a label for the task of interest from each synthetic future and estimate the probability of the outcome from how often it occurs across multiple simulated trajectories (Figure 1a).

While AR models have reported strong performance across many clinical prediction tasks (Waxler et al., 2025; Renc et al., 2024; 2025; Rajamohan et al., 2025; Shmatko et al., 2025; McDermott, 2025), this design has three critical problems. First, simulation is extremely expensive. To make a single prediction, an AR model must simulate many trajectories, each of which requires generating

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

many medical observations. As a result, *inference alone* over even small datasets can take thousands of GPU hours to complete (Solo et al., 2026). Second, these models are statistically noisy, and in particular struggle to predict rare outcomes. Concretely, autoregressive models estimate event probabilities by counting occurrences across K sampled trajectories. For rare events with probability $p \ll 1$, the expected number of trajectories containing the event is Kp, which may be very small even for large K. As a result, sample-driven estimates of the probability of these outcomes occurring have high variance and low sensitivity (Waxler et al., 2025; Renc et al., 2025). Third, AR models are not promptable: the task of interest is not an input to the model. Predictions for a given task are therefore obtained only indirectly, by post-processing simulations, rather than by querying the model for that task directly.

Here, we address these shortcomings by leveraging the insight that clinical prediction tasks are expressible in a parametric language. Given any structured query at inference, our model, EveryQuery, takes this query as an input prompt along with the patient's historical record and estimates the corresponding future outcome. Whereas autoregressive pretraining operates on medical events, EveryQuery learns a shared structure over medical events and a query language for clinical prediction tasks. As a result, EveryQuery is promptable, and can make direct predictions for clinically meaningful tasks without simulation (Figure 1b). We validate our approach on three datasets, including MIMIC-IV (Johnson et al., 2023), NWICU (Moukheiber et al., 2024), and a large academic medical center hereafter referred to as "AMC", and make the following contributions:

1. **A new pretraining approach for clinical prediction on structured EHR data**: Pretraining jointly over medical events and a query language for clinical tasks is a viable alternative to generative simulation, opening a new direction for EHR foundation models.
2. **A foundation model with superior performance across diverse clinical tasks**: EveryQuery outperforms an autoregressive baseline on both classification and time-to-event prediction tasks. Across clinically meaningful tasks, EveryQuery achieves win rates of 100% on MIMIC-IV, 94% on AMC, and 66.7% on NWICU (Wilcoxon signed-rank $p < 10^{-4}$ on MIMIC-IV and AMC; not significant on NWICU). EveryQuery demonstrates significant performance gains on rare outcomes.
3. **Efficient and tractable inference compared to autoregressive models**: As EveryQuery can be prompted with tasks at inference, outcomes can be estimated directly from the model. EveryQuery requires 582$\times$ lower compute than AR modeling and simulation-based inference.

2 Background

Structured clinical data. A patient's structured electronic health record (EHR) is a longitudinal sequence of discrete clinical events sourced from a particular medical institution. Each observation can be represented as a triple of a timestamp, a categorical medical "code" (e.g., a diagnostic code or laboratory test), and an optional numeric value (Arnrich et al., 2024). Increasingly, however, models discretize all aspects of these observations into a sequence of purely categorical "tokens". In this scheme, continuous numeric values (e.g., laboratory test results) and time deltas between observations are represented through quantized categorical tokens. As such, the entire EHR dataset is represented as ordinal sequences over a unified vocabulary of tokens which we will denote $\mathcal{V}$ (Renc et al., 2024; Wornow et al., 2025; Guo et al., 2026; Lee et al., 2026). We write a patient's tokenized record as $\mathbf{x} = (x_1, \ldots, x_L)$ with $x_i \in \mathcal{V}$, where elapsed time t_j at any position j is recovered by accumulating the time-delta tokens up to j. Because $\mathcal{V}$ is highly specific to the medical institution from which the EHR dataset is derived, foundation models trained over these token sequences are generally trained and evaluated within a single dataset and cannot be transferred to a new dataset without additional work to harmonize vocabularies. Most existing EHR foundation models follow this practice (Waxler et al., 2025; Renc et al., 2024; 2025).

Prediction tasks on EHR data. EHR data is used to predict diverse tasks of significant clinical interest, such as whether a patient will be readmitted to the hospital within 30 days (Tang et al., 2023), whether a patient is at imminent risk of mortality (Pang et al., 2025; Harutyunyan et al., 2019), or how long it will take until the patient first develops a disease (Shmatko et al., 2025; Steinberg et al., 2024). It is well established that clinical prediction tasks can be expressed in a deterministic, parametric form, through, e.g., SQL queries or ACES (Xu et al., 2024) configuration files. Such task definitions typically share a common structure: an outcome of interest, a prediction window, and optionally

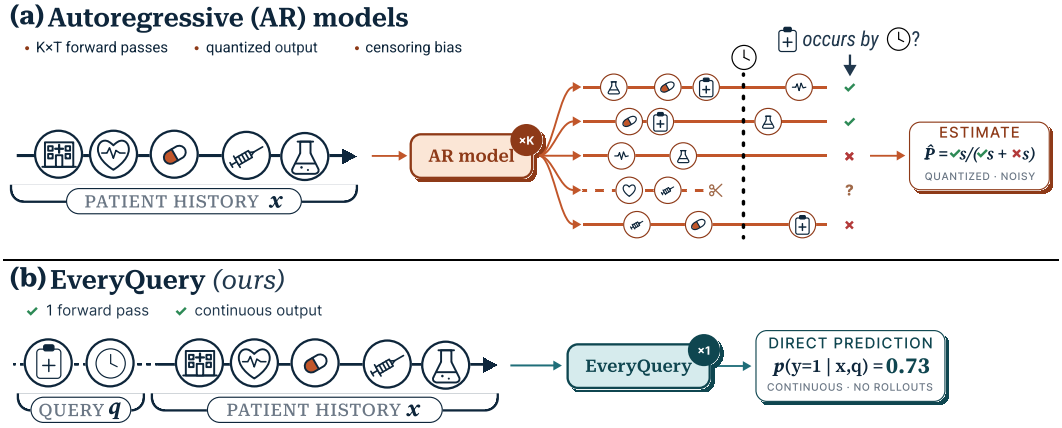

Figure 1: **Overview of EveryQuery.** Both models answer a query $\mathbf{q}$ during inference (§3.1), estimating whether a code occurs before a given time, for a patient history $\mathbf{x}$. **(a)** An autoregressive model generates K future trajectories of up to T events each, and estimates the probability as the fraction of trajectories in which the code occurs. This yields quantized, high-variance estimates. **(b)** EveryQuery takes $\mathbf{q}$ as input alongside $\mathbf{x}$ and predicts $p(y = 1 \mid \mathbf{x}, \mathbf{q})$ directly.

events that condition on or bound that window (e.g., predicting in-hospital mortality only for patients admitted to the ICU, and only until discharge). This structure supports many forms of prediction, including binary classification, joint prediction of several outcomes, and time-to-event estimation.

Autoregressive EHR foundation models. AR models ingest a patient's EHR data as a tokenized sequence and predict the next token, i.e., they model $p_\theta(x_i \mid x_{<i})$ through iterative multiclass classification over $\mathcal{V}$ at each position. This pretraining regime makes AR models capable of generating arbitrary token sequences, or "simulated futures" for the patient. Under a simple Monte Carlo approach, predictions on clinical tasks are made by extracting the target label from each synthetic future and aggregating these labels into an empirical distribution (Figure 1a) (Waxler et al., 2025; Renc et al., 2024; 2025). Advanced methods, such as importance sampling (Solo et al., 2026), improve efficiency, but prediction still requires hundreds to thousands of inference passes.

Task-specific and representation-learning models. Several other families of models are trained on structured EHR data, but none can make predictions across a diverse set of clinical tasks arbitrarily defined at inference time. Supervised models such as XGBoost (Chen & Guestrin, 2016) are trained for a single, pre-specified task and are unable to generalize to new tasks. Multitask models such as MOTOR (Steinberg et al., 2024) and M3H (Bertsimas & Ma, 2024) predict a large, but fixed and incomplete set of tasks through dedicated output heads chosen before training, and are unable to adapt to tasks outside this set. Representation-learning models that pretrain bidirectional or encoder-decoder representations on EHR data (Rasmy et al., 2021; Pang et al., 2021; Wornow et al., 2023; Yang et al., 2023; Steinberg et al., 2021) produce patient embeddings that require fine-tuning or linear probing for each new task, and therefore cannot make predictions for tasks specified at inference without task-specific labeled data. Among existing approaches, only AR models can address diverse clinical prediction tasks, from binary classification to time-to-event estimation, with a single pretrained model at inference time.

3 The EveryQuery Method

Driven by the insight that clinical prediction tasks can be expressed in a simple, structured form, we propose an alternative to autoregressive pretraining that sidesteps trajectory sampling entirely. EveryQuery learns representations of the patient history $\mathbf{x}$ conditioned on a query task $\mathbf{q}$ by taking both as input and predicting the answer directly, $p_\theta(y \mid \mathbf{x}, \mathbf{q})$ (Figure 1b). During pretraining, we construct pairs of patient histories and synthetic tasks, where tasks are randomly sampled from a continuous distribution over a query language. This query language distribution is specified during

Table 1: **Example prompts in the query language.**

Task	**Conditional Clauses** (q_i, a_i)	**Target** q_n
In-hospital mortality	–	`(death, (0, discharge))`
30-day readmission	–	`(admission, (discharge, 30d))`
Post-discharge mortality	`((death, (0, discharge)), 0)`	`(death, (discharge, 90d))`

pretraining in a manner that is completely agnostic to inference tasks. EveryQuery has the same "API" as autoregressive models, where it is pre-trained over an unlabeled EHR dataset in a task-agnostic manner, and can then answer diverse prediction tasks at inference time.

3.1 Query Language

Structured queries. EveryQuery leverages the natural parametric structure of clinical prediction tasks (Xu et al., 2024). We define the basic unit of our query language as a *structured query* $q_i = (c_i, w_i)$, which asks whether a clinical event $c_i \in \mathcal{V}$ occurs within a future window $w_i = (s_i, e_i)$. Each window boundary is either a duration in $\mathbb{R}_+$ or a clinical event in $\mathcal{V}$, which resolves to the time of that event's next occurrence. A duration-specified start is defined relative to the prediction time t_L, whereas a duration-specified end is defined relative to the resolved window start. With $\tau(s_i)$ and $\tau(e_i)$ denoting the resolved start and end times, the answer for patient $\mathbf{x}$ is $\mathbb{1}_{q_i}(\mathbf{x}) = \mathbb{1}[\exists\, j > L : x_j = c_i,\ \tau(s_i) < t_j \leq \tau(e_i)]$. Event-bounded windows allow structured queries to follow the course of care rather than fixed durations. Table 1 lists examples.

Conditional clauses. A structured query can be extended with *conditional clauses*, which enable complex questions about future events, in which a target structured query q_n is asked given that other structured queries q_i do $(a_i = 1)$ or do not $(a_i = 0)$ occur in the patient's future. We formalize a prompt input $\mathbf{q} = ((q_0, a_0), \ldots, (q_{n-1}, a_{n-1}), q_n)$ as a target structured query q_n asked conditional on the clauses $(q_i, a_i)_{i<n}$, where each clause specifies the observed outcome $a_i \in \{0, 1\}$ of a preceding structured query q_i. The answer to $\mathbf{q}$ is defined only for patients whose futures satisfy every condition, $y(\mathbf{x}, \mathbf{q}) = \mathbb{1}_{q_n}(\mathbf{x})$ if $\mathbb{1}_{q_i}(\mathbf{x}) = a_i \ \ \forall i < n$, and EveryQuery estimates $p_\theta(y = 1 \mid \mathbf{x}, \mathbf{q}) \approx P\big(\mathbb{1}_{q_n}(\mathbf{x}) = 1 \mid \mathbf{x},\ \mathbb{1}_{q_i}(\mathbf{x}) = a_i\ \forall i < n\big)$. With $n = 0$, the prompt is a single structured query with no conditional clauses.

Clinical ontology. Clinical questions are often asked about groups of codes rather than a single code. For example, a clinician may want to know whether a patient will be discharged at all, not whether they will be discharged home or to a skilled nursing facility, which are separate codes in the vocabulary. EveryQuery supports such questions by arranging the codes of the dataset vocabulary into a hierarchy, in which a parent code such as `discharge` stands for all of its more specific descendants. Every parent code is added to the vocabulary $\mathcal{V}$ with its own embedding, and a structured query about a parent code is answered positively if any of its descendants occurs in the window.

3.2 Model

Input sequence. Given a patient history $\mathbf{x}$ up to the prediction time t_L and a prompt $\mathbf{q}$ with conditional clauses $(q_i, a_i)_{i<n}$ and target structured query q_n (§3.1), EveryQuery processes a single sequence

$$[\, x_1, \ldots, x_L,\ w_0, c_0, a_0,\ \ldots,\ w_{n-1}, c_{n-1}, a_{n-1},\ w_n \,],$$

in which each conditional clause contributes a window token w_i, a code token c_i and an answer token a_i, and the target structured query $q_n = (c_n, w_n)$ contributes only its window token w_n. Patient history is truncated to the most recent 256 events. EveryQuery uses a Llama-style decoder-only transformer (Touvron et al., 2023) trained from scratch, with standard causal attention: each window token w_i attends to the full patient history, to every earlier conditional clause, and to itself, but never to the answer for its own window.

Time and position. EveryQuery encodes time through its position embeddings rather than via time-delta tokens. Time-delta tokens are removed from the input after their durations are accumulated,

which shortens sequences so that more clinical events fit in the context window. Each preserved event's rotary position embedding is computed from its elapsed time in hours rather than its index in the sequence. All query tokens are assigned the prediction time t_L, anchoring them to the end of the patient context. A nonzero window start is expressed through its window token rather than position.

Token embeddings. Every input token is embedded as the sum of two learned vectors: a *content* vector, which encodes what the token says, and a *token-type* vector denoted $\mathbf{t}_{\text{type}}$, which encodes what kind of token it is. There are four token types: patient event, window, clause code and clause answer. Query tokens additionally receive a *block* vector $\mathbf{b}_i$, which identifies the structured query q_i they belong to. The resulting embeddings are:

Token	Embedding	Content vector
Patient event x_j	$E[x_j] + \mathbf{t}_{\text{patient}}$	code embedding
Window w_i	$\mathbf{u}(s_i) + \mathbf{v}(e_i) + \mathbf{t}_{\text{window}} + \mathbf{b}_i$	start and end embeddings
Clause code c_i	$E[c_i] + \mathbf{t}_{\text{code}} + \mathbf{b}_i$	code embedding
Clause answer a_i	$A[a_i] + \mathbf{t}_{\text{answer}} + \mathbf{b}_i$	answer embedding

All codes, whether patient events, clause codes or window boundaries, are embedded with a single shared table E. Numeric values such as laboratory results are binned into the code during preprocessing. Answers and durations have embeddings of their own: answers use a learned two-entry table A, and durations use the MLPs described below. When the clinical ontology (§3.1) is used, each row of E is replaced by a normalized average of that code's embedding and its ancestors' embeddings, with weights that decay with distance, so that related codes, including rare ones, share information.

Window boundaries. The start embedding $\mathbf{u}(s_i)$ and end embedding $\mathbf{v}(e_i)$ depend on how each boundary is specified. An event boundary is embedded as the event's code embedding plus a learned marker to distinguish start from end, $\mathbf{u}(s_i) = E[s_i] + \mathbf{m}_{\text{start}}$ or $\mathbf{v}(e_i) = E[e_i] + \mathbf{m}_{\text{end}}$. Temporal start and end boundaries are converted to years and processed by separate MLPs.

Readout. The target structured query $q_n = (c_n, w_n)$ asks whether code c_n occurs in window w_n. The window w_n is given to the model as an input token, but the code c_n is not. Instead, the model reads out its answer by comparing its output at w_n, denoted $\mathbf{h}_n$, with the embedding of c_n:

$$p_\theta(y = 1 \mid \mathbf{x}, \mathbf{q}) = \sigma\big(\mathbf{h}_n^\top E[c_n] + \beta_{c_n}\big),$$

where E is the same code embedding table used for the inputs and β_c is a learned per-code bias.

3.3 PRETRAINING

Objective. The pretraining objective is to predict the answer to every structured query in a training prompt. Because $\mathbf{h}_n$ does not depend on c_n, the readout above can score every code $c \in \mathcal{V}$ at once. We apply it at every window token w_i, not only the target, so that each window in a training prompt supervises $|\mathcal{V}|$ structured queries that share its window and preceding clauses, or $(n+1)|\mathcal{V}|$ structured queries per prompt. The model is trained with binary cross-entropy averaged over all windows and codes, with the clause answers a_i teacher-forced from the patient's true future. EveryQuery therefore learns $p_\theta\big(\mathbb{1}_{(c,w_i)}(\mathbf{x}) = 1 \mid \mathbf{x}, w_0, \ldots, w_i, (c_j, a_j)_{j<i}\big)$ for every code c and every window i. Whereas autoregressive pretraining performs multiclass classification over $\mathcal{V}$, predicting which one code comes next, EveryQuery solves $|\mathcal{V}|$ simultaneous binary classification problems at each window. Supervising this many queries per prompt reduces the variance of the gradient.

Training. We construct 5 million training contexts. For each, we draw a patient uniformly at random, then sample a prediction time uniformly after excluding times with fewer than 50 distinct preceding timestamps, and sample a query with five windows. Each window starts at the prediction time (40%), after a delay drawn log-uniformly from 1 to 180 days (40%), or at an event (20%), and ends either at an event (50%) or after a duration drawn log-uniformly from 1 to 1,826 days (50%). Boundary events and clause codes are drawn uniformly over all codes, including ontology ancestors, independently of the patient's data. Further optimization details are in Appendix B.

4 EXPERIMENTS

Datasets. We train and evaluate on three EHR datasets, each preprocessed into the MEDS format (McDermott et al., 2026): MIMIC-IV (v2.2) (Johnson et al., 2023), NWICU (Moukheiber et al., 2024), and a private EHR dataset from a large academic medical center, hereafter AMC. The three datasets vary substantially in population, care setting, and scale, with population sizes of 240K, 24K, and 1.9M patients after preprocessing. Data splits are described in Appendix A. All results are reported on the held-out test split of each dataset. Following convention, we train one EveryQuery and one autoregressive baseline on each dataset.

Baseline. We compare against MEDS-EIC-AR (McDermott, 2025)[1], an open-source "everything-is-code" autoregressive model that fuses each clinical code with its decile-binned value into a single token, a design supported by recent tokenization benchmarks for structured EHR foundation models (Lee et al., 2026; Guo et al., 2026). MEDS-EIC-AR is one of the few released AR foundation models that can be trained on any MEDS dataset, and it has been shown to be effective for pretraining, fine-tuning and representation learning (Pollet & McDermott, 2026). The model has 102M parameters, uses a max sequence length of 1024 tokens, of which at most 256 are reserved for the input patient context and the remainder for autoregressive generation via next-token prediction. Both EveryQuery and MEDS-EIC-AR therefore operate on the same effective patient history context of 256 tokens at inference time, though MEDS-EIC-AR benefits from longer contexts during training. At inference, we sample $K = 20$ future trajectories per patient, as is typical (Renc et al., 2025), and estimate each task's probability from the labels extracted from these trajectories.

Evaluation. We evaluate on diverse classification tasks, each anchored at a clinically significant prediction time and optionally defined over a cohort of interest. The 38 and 27 tasks on the MIMIC-IV and NWICU datasets are anchored at 24 hours after ICU admission. The 100 tasks on the AMC dataset in the longitudinal care setting are anchored at outpatient visits, and some are defined over type 2 diabetes and heart failure cohorts. Tasks cover a wide range of clinical events described in Figure 3. Where appropriate, tasks are conditioned on the patient being neither censored nor deceased within the window, and some add other task-specific clauses.

To further investigate tasks that are not directly expressible in the query language, we evaluate whether EveryQuery can recover survival curves for events of interest, and report comparisons to Kaplan–Meier estimates. Given the time T to the first occurrence of a target code c, with records censored at their last event, the survival function is $S(t) = P(T > t)$. EveryQuery estimates the survival function at each time t from three structured queries: the probability that c occurs by t, $F(t)$; the probability that the record ends by t, $Q(t)$; and the probability that c occurs by t given that the record remains open, $U(t)$. Together these give the probability of being event-free and still at risk, $R(t) = (1 - Q(t))(1 - U(t))$, and hence the hazard $\lambda(t) = F'(t)/R(t)$, the same quantity that Kaplan–Meier estimates. The survival curve then follows as $S(t) = \exp\big(-\int_0^t \lambda(s)\, ds\big)$. The autoregressive baseline instead pools its K trajectories into a per-patient Kaplan–Meier curve. We report Harrell's C, and list target codes and prediction horizons for each dataset in Appendix D.

5 RESULTS

5.1 EVERYQUERY OUTPERFORMS AR BASELINE AT DIVERSE CLINICAL PREDICTION TASKS

EveryQuery outperforms the autoregressive baseline on both classification and time-to-event tasks, with large gains on MIMIC-IV and AMC and comparable performance on NWICU (Figure 2).

Classification. On the clinically relevant classification tasks, EveryQuery achieves a higher AUROC than the autoregressive baseline on 100% of tasks on MIMIC-IV, 66.7% on NWICU and 94% on AMC (Figure 2a). The win rates on MIMIC-IV and AMC are significantly above chance (Wilcoxon signed-rank $p < 10^{-4}$). Averaged over tasks, EveryQuery reaches AUROCs of 84.9%, 86.6% and 83.6% on MIMIC-IV, NWICU and AMC. Its mean AUROC gain over the autoregressive baseline is $+14.3\%$ (95% CI $[11.7\%, 17.0\%]$) on MIMIC-IV and $+12.7\%$ ($[10.5\%, 14.8\%]$) on AMC. On

[1] https://github.com/mmcdermott/MEDS_EIC_AR

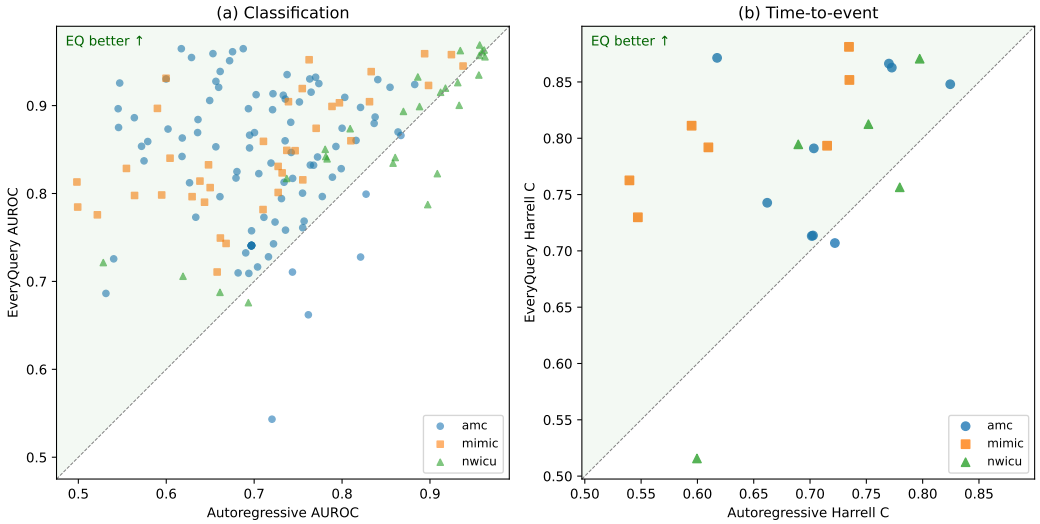

Figure 2: **EveryQuery outperforms the autoregressive baseline across clinical prediction tasks.** Each point is a task, comparing EveryQuery with the AR baseline on AUROC for classification tasks (a) and Harrell's C for time-to-event tasks (b). Points above the diagonal favor EveryQuery.

NWICU, the mean gain is smaller and not significant ($+1.6\%$, $[-0.4\%, 3.8\%]$), consistent with the lower win rate on this dataset. Per-task AUROCs are listed in Appendix C.

Time-to-event. Although time-to-event tasks are not directly expressible in the query language, EveryQuery still outperforms the autoregressive baseline. EveryQuery reaches a mean Harrell's C of 80.3%, 75.0% and 79.1% on MIMIC-IV, NWICU and AMC. EveryQuery achieves mean per-task win margins over the autoregressive baseline of $+16.4\%$ (95% CI $[11.5\%, 21.3\%]$), $+2.6\%$ ($[-7.0\%, 12.3\%]$) and $+7.1\%$ ($[0.9\%, 13.3\%]$) (Figure 2b). The margins on MIMIC-IV and AMC are significant in Wilcoxon signed-rank test, while NWICU is again comparable. Survival curves for all tasks are shown in Appendix D.

5.2 EveryQuery is strongest when making predictions for rare events

EveryQuery performs best on rare events, which are often the outcomes of greatest clinical interest and the hardest for simulation-based models to estimate (Figure 3). Across classification tasks on MIMIC-IV, NWICU and AMC, EveryQuery's AUROC is negatively correlated with task prevalence on all three datasets (Spearman $\rho = -0.40$, $p = 0.014$ on MIMIC-IV; $\rho = -0.59$, $p = 0.001$ on NWICU; $\rho = -0.38$, $p = 10^{-4}$ on AMC), so EveryQuery has better discrimination on rarer outcomes.

As a result, EveryQuery's advantage over the autoregressive baseline grows as events become rarer. The win margin (ΔAUC) is negatively correlated with prevalence on MIMIC-IV ($\rho = -0.35$, $p = 0.03$) and AMC ($\rho = -0.61$, $p = 10^{-11}$). The autoregressive baseline does not share EveryQuery's advantage on rare events in these datasets: its AUROC is uncorrelated with prevalence on MIMIC-IV ($\rho = +0.11$, $p = 0.52$) and positively correlated with prevalence on AMC ($\rho = +0.45$, $p = 2 \times 10^{-6}$), meaning it performs worse on rarer events. On NWICU, the correlation between win margin and prevalence is not significant ($\rho = +0.23$, $p = 0.24$).

5.3 EveryQuery is more efficient at inference than autoregressive modeling

EveryQuery requires a single deterministic forward pass per (patient, query) pair. The autoregressive baseline requires generating $K = 20$ full future trajectories, each of which has hundreds of events on average, per patient. To compare inference efficiency independently of hardware, we measure the number of floating point operations (FLOPs) required per (patient, query) for each model. EveryQuery requires only 7.76 giga FLOPs per query, compared with an average of 4.52 tera FLOPs

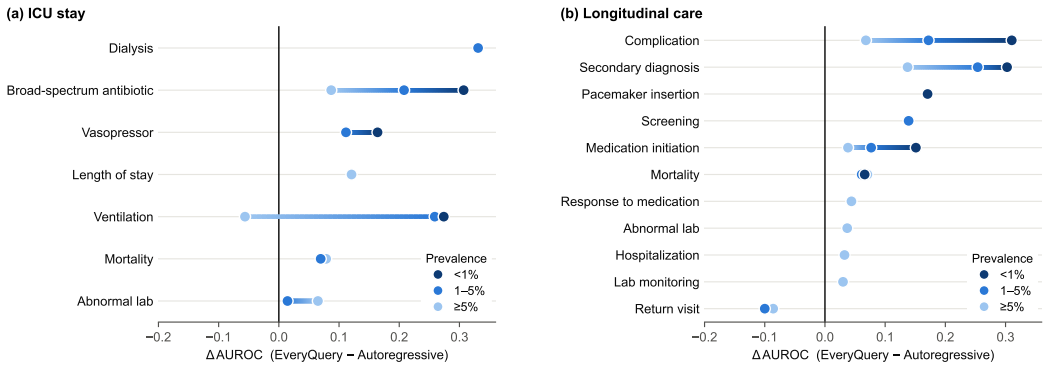

Figure 3: **EveryQuery's advantage over the autoregressive baseline is largest for rare events.** Each row is a category of clinical prediction tasks, and each dot shows the mean ΔAUROC for the tasks in that category grouped by prevalence. ICU tasks **(a)** are evaluated on MIMIC-IV and NWICU; and longitudinal care tasks **(b)** on AMC. EveryQuery's gains grow as outcomes become rarer.

for the optimized autoregressive baseline, corresponding to approximately 582× lower inference compute required for EveryQuery. Notably, this comparison already assumes a heavily optimized autoregressive inference implementation that reuses the prefilled patient context across all ($K = 20$) generated trajectories. Without this level of optimization, using naïve autoregressive generation directly, EveryQuery achieved approximately 5,500 to 10,300 times faster queries-per-second during inference.

5.4 EveryQuery's predictions respect temporal and hierarchical structure

EveryQuery is not constrained to give consistent answers to related queries, but the structure of the query language implies that some of its answers should agree. Consider the prediction horizon, where the probability that a code occurs should be monotonically increasing in the length of the window given that the windows have the same start time. We therefore checked whether EveryQuery's predicted probabilities are monotonically increasing as the horizon grows. With a tolerance of 10^{-3} in predicted probability, EveryQuery's predictions are monotonically increasing across all consecutive horizons for 98.5% of (patient, code) pairs on MIMIC-IV and 87.5% on NWICU. EveryQuery thus learns a relatively coherent notion of time without any architectural constraint.

The clinical ontology implies a similar constraint across codes. An ancestor code occurs whenever any of its descendants is observed (§3.1), so for the same patient and window, the probability of the ancestor should be at least as high as the probability of each of its descendants. We tested this for five ontology subtrees in MIMIC-IV and NWICU by comparing, for each patient and window, the predicted probability of the ancestor with that of each of its descendants. The ancestor received a probability at least as high in 98.4% of these comparisons on MIMIC-IV and 99.6% on NWICU. EveryQuery therefore also learns the hierarchical structure of the ontology.

5.5 EveryQuery's performance is robust across the query language

Since the clinically relevant tasks above were manually curated, we assess whether EveryQuery's performance remains robust across the query language. We repeat the evaluations above on randomly sampled tasks, following prior work showing that relative performance on randomly sampled clinical prediction tasks tracks relative performance on expert-authored tasks (Pollet & McDermott, 2026). For each dataset, we sample 1,000 prompts and reject tasks with fewer than 40 positive cases following (Pollet & McDermott, 2026).

EveryQuery remains robust to choice of tasks. EveryQuery achieves a higher AUROC than the autoregressive baseline on 100% of tasks on MIMIC-IV, 95% on NWICU and 100% on AMC, and its macro-averaged AUROC is significantly higher on every dataset (Wilcoxon signed-rank $p < 10^{-4}$). As in §5.2, EveryQuery's win margin is also significantly negatively correlated with task prevalence on all three datasets with Spearman $\rho = -0.50$, -0.40, and -0.59 for MIMIC-IV, NWICU, and

AMC, respectively all with $p < 10^{-4}$ (Appendix Figure 7). EveryQuery's advantages are therefore not specific to the tasks we selected, but hold across the query language.

6 Discussion and Conclusion

Our evaluations of EveryQuery in Section 5 demonstrate that learning a shared structure over medical events and a query language for clinical tasks is an advantageous alternative to autoregressive generation for the development of EHR foundation models. In particular, across all three datasets we study, EveryQuery shows clear advantages in performance and computational efficiency over a compelling AR model baseline across both binary classification and survival modeling tasks. In this section, we offer further commentary on these results, including specific limitations of EveryQuery that should be investigated and future work.

Limitations and future work. EveryQuery has several key limitations that should be explored in future work. First and most critically, while the query language we study here is quite expressive and permits EveryQuery to directly model a wide variety of clinical prediction tasks of different types, it is not truly universal. A key area of future work, thus, is to explore extending EveryQuery's query language to cover all possible tasks over structured EHR data, such as by using a raw text expression as a query during pre-training. Beyond the query language specificity, EveryQuery's pre-training program also samples from uniform distributions over synthetic queries and patient contexts. In the real world, only a subset of these pairs are actually of clinical interest, and training efficiency may be able to be improved through more intelligent sampling methods leveraging ideas from curriculum learning. While we selected MEDS-EIC-AR for its many strengths, comparisons with other dataset-specific EHR foundation models, such as ETHOS (Renc et al., 2024), would further strengthen our conclusions. Finally, like most foundation models over EHR data, EveryQuery can only be used on datasets sharing the same code vocabulary $\mathcal{V}$ as was seen during training. Methods to account for this, such as using universal, text-based encoders (Hur et al., 2023), could be explored to help this.

Conclusion. EveryQuery offers an alternative to autoregressive generation for EHR foundation models. Across diverse clinical prediction tasks, it reliably outperforms a competitive autoregressive baseline in both performance and efficiency, with the largest gains on rare events, while using far less inference compute. This approach offers a compelling new direction for EHR foundation model research and further exposes new opportunities for research in other longitudinal domains where tasks can be expressed parametrically.

AI Use Statement

We used generative AI throughout, including to implement methods, help formalize the query language, perform limited experimental loop supervision, draft the paper, help make figures, summarize related literature, and format references. We did not use AI to interpret results or generate synthetic datasets. Proofs, translation, and qualitative or thematic data analysis are not applicable. We reviewed all AI-assisted work, including code, reported numbers, and citations. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics Statement

This work uses de-identified electronic health record data. MIMIC-IV and NWICU were accessed using credentials. AMC data use an IRB-approved protocol (details withheld for anonymous review).

Reproducibility Statement

We provide complete details about data, preprocessing, model architecture, pretraining, and optimization. EveryQuery is MEDS (McDermott et al., 2026) compliant and can be trained on other MEDS datasets without modification. On publication, all our code will be released publicly.

REFERENCES

Bert Arnrich, Edward Choi, Jason Alan Fries, Matthew B A McDermott, Jungwoo Oh, Tom Pollard, Nigam Shah, Ethan Steinberg, Michael Wornow, and Robin van de Water. Medical event data standard (MEDS): Facilitating machine learning for health. *OpenReview ICLR Workshop*, pp. 03–08, March 2024.

Dimitris Bertsimas and Yu Ma. M3H: Multimodal multitask machine learning for healthcare. *arXiv [cs.LG]*, April 2024.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, New York, NY, USA, 13 August 2016. ACM. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

Lin Lawrence Guo, Santiago Eduardo Arciniegas, Joseph Jihyung Lee, Adam Paul Yan, George Tomlinson, Jason Fries, and Lillian Sung. Tokenization tradeoffs in structured EHR foundation models. *arXiv [cs.LG]*, 2026. URL http://arxiv.org/abs/2603.15644.

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

Kyunghoon Hur, Jungwoo Oh, Junu Kim, Jiyoun Kim, Min Jae Lee, Eunbyeol Cho, Seong-Eun Moon, Young-Hak Kim, Louis Atallah, and Edward Choi. Genhpf: General healthcare predictive framework for multi-task multi-source learning. *IEEE Journal of Biomedical and Health Informatics*, 28(1):502–513, 2023.

Alistair E W Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-Wei H Lehman, Leo A Celi, and Roger G Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10:1, 2023. URL http://dx.doi.org/10.1038/s41597-022-01899-x.

Inhyeok Lee, Luke Solo, Michael C Burkhart, Bashar Ramadan, William F Parker, and Brett K Beaulieu-Jones. Representation before training: A fixed-budget benchmark for generative medical event models. *arXiv [cs.LG]*, 2026. URL http://dx.doi.org/10.48550/arXiv.2604.16775.

Matthew McDermott. MEDS "everything-is-code" autoregressive model, 2025.

Matthew B A McDermott, Ethan Steinberg, Jason A Fries, Robin P van de Water, Chao Pang, Patrick Rockenschaub, Pawel Renc, Jungwoo Oh, Kamilė Stankevičiūtė, Justin Xu, Tom J Pollard, Nassim Oufattole, Michael Wornow, Teya S Bergamaschi, Hyewon Jeong, Simon A Lee, Vincent Jeanselme, Kiril V Klein, Mikkel Odgaard, Maria E Montgomery, Arkadiusz Sitek, Mads Nielsen, Jeffrey N Chiang, Noa Dagan, Isaac Kohane, Shalmali Joshi, Edward Choi, and Nigam H Shah. MEDS - an emerging data standard and ecosystem for health AI research. *NEJM AI*, 3(6), 28 June 2026. ISSN 2836-9386. doi: 10.1056/AIra2501253. URL http://dx.doi.org/10.1056/AIra2501253.

Dana Moukheiber, William Temps, Bhadrappa Molgi, Yikuan Li, Alice Lu, Prasanth Nannapaneni, Abdulrahman Chahin, Sicheng Hao, Felipe Torres Fabregas, Leo Anthony Celi, Adrian Wong, Maxwell Lloyd, Xavier Borrat Frigola, Hyung-Chul Lee, Daniel Schneider, Tom Pollard, Yuan Luo, Abel Kho, and Roger Mark. Northwestern ICU (NWICU) database. *PhysioNet*, November 2024. doi: 10.13026/s84w-1829. URL https://doi.org/10.13026/s84w-1829. Version 0.1.0.

Chao Pang, Xinzhuo Jiang, Krishna S Kalluri, Matthew Spotnitz, Ruijun Chen, Adler Perotte, and Karthik Natarajan. CEHR-BERT: Incorporating temporal information from structured EHR data to improve prediction tasks. *arXiv [cs.LG]*, November 2021.

Chao Pang, Vincent Jeanselme, Young Sang Choi, Xinzhuo Jiang, Zilin Jing, Aparajita Kashyap, Yuta Kobayashi, Yanwei Li, Florent Pollet, Karthik Natarajan, and Shalmali Joshi. FoMoH: A clinically meaningful foundation model evaluation for structured electronic health records. *arXiv [cs.LG]*, May 2025.

Florent Pollet and Matthew McDermott. Rethinking how we evaluate methodological progress in health AI. *arXiv preprint arXiv:2609.18134*, 2026.

Haresh Rengaraj Rajamohan, Xiang Gao, Weicheng Zhu, Shih-Lun Huang, Long Chen, Kyunghyun Cho, Cem M Deniz, and Narges Razavian. Foundation models for clinical records at health system scale. *arXiv [cs.LG]*, July 2025.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digit. Med.*, 4(1):86, May 2021.

Pawel Renc, Yugang Jia, Anthony E Samir, Jaroslaw Was, Quanzheng Li, David W Bates, and Arkadiusz Sitek. Zero shot health trajectory prediction using transformer. *npj Digital Medicine*, 7 (1):1–10, September 2024.

Pawel Renc, Michal K Grzeszczyk, Nassim Oufattole, Deirdre Goode, Yugang Jia, Szymon Bieganski, Matthew B A McDermott, Jaroslaw Was, Anthony E Samir, Jonathan W Cunningham, David W Bates, and Arkadiusz Sitek. Foundation model of electronic medical records for adaptive risk estimation. *arXiv [cs.LG]*, February 2025.

Artem Shmatko, Alexander Wolfgang Jung, Kumar Gaurav, Søren Brunak, Laust Hvas Mortensen, Ewan Birney, Tom Fitzgerald, and Moritz Gerstung. Learning the natural history of human disease with generative transformers. *Nature*, 647(8088):248–256, November 2025.

Luke Solo, Matthew B A McDermott, William F Parker, Bashar Ramadan, Michael C Burkhart, and Brett K Beaulieu-Jones. Efficient variance-reduced estimation from generative EHR models: The SCOPE and REACH estimators. *arXiv [stat.ML]*, February 2026.

Ethan Steinberg, Ken Jung, Jason A Fries, Conor K Corbin, Stephen R Pfohl, and Nigam H Shah. Language models are an effective representation learning technique for electronic health record data. *Journal of biomedical informatics*, 113:103637, 2021.

Ethan Steinberg, Jason Fries, Yizhe Xu, and Nigam Shah. MOTOR: A time-to-event foundation model for structured medical records. In *International Conference on Learning Representations*, 2024.

Siyi Tang, Amara Tariq, Jared A Dunnmon, Umesh Sharma, Praneetha Elugunti, Daniel L Rubin, Bhavik N Patel, and Imon Banerjee. Predicting 30-day all-cause hospital readmission using multimodal spatiotemporal graph neural networks. *IEEE Journal of Biomedical and Health Informatics*, 27(4):2071–2082, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Shane Waxler, Paul Blazek, Davis White, Daniel Sneider, Kevin Chung, Mani Nagarathnam, Patrick Williams, Hank Voeller, Karen Wong, Matthew Swanhorst, Sheng Zhang, Naoto Usuyama, Cliff Wong, Tristan Naumann, Hoifung Poon, Andrew Loza, Daniella Meeker, Seth Hain, and Rahul Shah. Generative medical event models improve with scale. *arXiv [cs.LG]*, August 2025.

Michael Wornow, Rahul Thapa, Ethan Steinberg, Jason A Fries, and Nigam H Shah. EHRSHOT: An EHR benchmark for few-shot evaluation of foundation models. *arXiv [cs.LG]*, July 2023.

Michael Wornow, Suhana Bedi, Miguel Angel Fuentes Hernandez, Ethan Steinberg, Jason Alan Fries, Christopher Re, Sanmi Koyejo, and Nigam Shah. Context clues: Evaluating long context models for clinical prediction tasks on EHR data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zg3ec1TdAP.

Justin Xu, Jack Gallifant, Alistair E W Johnson, and Matthew B A McDermott. ACES: Automatic cohort extraction system for event-stream datasets. *arXiv [cs.LG]*, June 2024.

Zhichao Yang, Avijit Mitra, Weisong Liu, Dan Berlowitz, and Hong Yu. TransformEHR: transformer-based encoder-decoder generative model to enhance prediction of disease outcomes using electronic health records. *Nat. Commun.*, 14(1):7857, November 2023.

A DATASETS

Table 2: Dataset statistics per split after preprocessing.

	Subjects				
Dataset	Train	Val	Test	Total Events	Vocab
MIMIC	200,773	25,059	15,059	779,733,629	11,476
AMC	1,651,030	291,538	27,019	2,250,292,868	21,289
NWICU	20,723	2,590	772	54,785,364	2,159

B ARCHITECTURE AND TRAINING DETAILS

Table 3 summarizes the training and architecture hyperparameters used for EveryQuery.

Table 3: **EveryQuery training and architecture hyperparameters.**

Hyperparameter	**Value**
Optimization	
Optimizer	AdamW
Learning rate	5×10^{-4}
Betas	$(0.9, 0.98)$
Epsilon	10^{-8}
Weight decay	0.05
LR schedule	Cosine with warmup
Warmup steps	5208
Training steps	104,167
Wall-clock time	135 mins
Batch size	96
Precision	bf-16 mixed
Architecture	
Backbone	Llama decoder-only (from scratch)
Hidden dimension	384
Transformer layers	6
Attention heads	6
Backbone parameters	$\sim$22M
Max sequence length	256
MLP dropout	0.1
Data	
Min. context times per prediction	50
Sequence sampling strategy	TO_END
Padding side	RIGHT
Batch mode	SM
Infrastructure	
GPU	1$\times$ NVIDIA L40S (48 GB)

C Results on Classification Tasks

Tables below report AUROC gaps for EveryQuery and the autoregressive baseline across all classification tasks, sorted by task prevalence. Tasks are sorted by descending prevalence, the better AUROC per task is in bold.

Table 4: AMC outpatient visit, general cohort ($n = 5$ tasks).

Task	Prev.	AUROC	
		EveryQuery	Autoregressive
`Visit/OP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d`	0.730	0.543	**0.720**
`Visit/ER within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d`	0.114	0.728	**0.821**
`MEDS_DEATH within 1825d`	0.028	**0.887**	0.838
`Visit/ER within 30d | no TIMELINE//END within 30d, no MEDS_DEATH within 30d`	0.025	0.662	**0.762**
`MEDS_DEATH within 365d`	0.010	**0.921**	0.855

Table 5: AMC outpatient visit, heart-failure codes ($n = 36$ tasks).

Task	Prev.	AUROC	
		EveryQuery	Autoregressive
`LOINC/2823-3//ANY within 90d | no TIMELINE//END within 90d, no MEDS_DEATH within 90d`	0.623	0.799	**0.827**
`LOINC/2823-3//UCUM/mmol/L// value_[5.0,inf) within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d`	0.354	**0.769**	0.757
`ICD10CM/I48.91 within 1095d | no TIMELINE//END within 1095d`	0.284	**0.828**	0.799
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1373473 within 30d`	0.222	**0.741**	0.697
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1545658 within 30d`	0.222	**0.741**	0.697
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1545664 within 30d`	0.222	**0.741**	0.697
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1545666 within 30d`	0.222	**0.741**	0.697
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1373469 within 30d`	0.222	**0.741**	0.697
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1486977 within 30d`	0.222	**0.741**	0.697
`Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1486981 within 30d`	0.222	**0.741**	0.697

continued on next page

Task	Prev.	EveryQuery	Autoregressive
Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1488569 within 30d	0.222	**0.741**	0.697
Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d, no RxNorm/1545668 within 30d	0.222	**0.741**	0.697
Visit/IP within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.222	**0.740**	0.697
MEDS_DEATH within 1825d	0.168	**0.773**	0.711
LOINC/2160-0//UCUM/mg/dL// value_[2.34,inf) within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.141	**0.924**	0.883
Visit/ER within 90d | no TIMELINE//END within 90d, no MEDS_DEATH within 90d	0.119	**0.728**	0.716
Visit/IP within 90d | no TIMELINE//END within 90d, no MEDS_DEATH within 90d	0.111	**0.758**	0.736
LOINC/33762-6//ANY within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.109	**0.909**	0.803
ICD10CM/N18.30 within 1095d | no TIMELINE//END within 1095d	0.109	**0.869**	0.700
ICD10CM/N18.3 within 1095d | no TIMELINE//END within 1095d	0.083	**0.912**	0.733
RxNorm/313096 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.068	**0.819**	0.789
ICD10CM/I48.0 within 1095d | no TIMELINE//END within 1095d	0.068	**0.915**	0.765
MEDS_DEATH within 365d	0.049	**0.817**	0.744
ICD10CM/I48.2 within 1095d | no TIMELINE//END within 1095d	0.021	**0.913**	0.721
RxNorm/200820 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.016	**0.852**	0.695
ICD10CM/I48.20 within 1095d | no TIMELINE//END within 1095d	0.014	**0.935**	0.737
RxNorm/351256 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.013	**0.904**	0.751
ICD10CM/I48.1 within 1095d | no TIMELINE//END within 1095d	0.011	**0.906**	0.649
ICD10CM/I48.19 within 1095d | no TIMELINE//END within 1095d	0.010	**0.939**	0.661
ICD10CM/N18.32 within 1095d | no TIMELINE//END within 1095d	0.010	**0.965**	0.617
ICD10PCS/0JH606Z within 730d | no TIMELINE//END within 730d	0.009	**0.726**	0.540
ICD10CM/N18.31 within 1095d | no TIMELINE//END within 1095d	0.009	**0.955**	0.629
RxNorm/198223 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.009	**0.825**	0.680
ICD10PCS/02HK3JZ within 730d | no TIMELINE//END within 730d	0.008	**0.686**	0.531

continued on next page

Task	Prev.	EveryQuery	Autoregressive
ICD10CM/I48.11 within 1095d | no TIMELINE//END within 1095d	0.007	**0.961**	0.675
ICD10CM/I48.21 within 1095d | no TIMELINE//END within 1095d	0.006	**0.951**	0.672

Table 6: AMC outpatient visit, type 2 diabetes codes ($n = 59$ tasks).

Task	Prev.	AUROC	
		EveryQuery	Autoregressive
LOINC/4548-4//ANY within 180d | no TIMELINE//END within 180d, no MEDS_DEATH within 180d	0.301	0.866	**0.867**
ICD10CM/I50.814 within 1095d | no TIMELINE//END within 1095d	0.151	**0.898**	0.821
LOINC/2160-0//UCUM/mg/dL//value_[2.34,inf) within 730d | no TIMELINE//END within 730d	0.137	**0.930**	0.841
MEDS_DEATH within 1825d	0.121	**0.813**	0.734
ICD10CM/E11.319 within 1095d | no TIMELINE//END within 1095d	0.116	**0.832**	0.764
LOINC/14959-1//ANY within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.107	**0.880**	0.837
LOINC/4548-4//UCUM/%//value_[8.9,inf) within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.094	**0.870**	0.864
ICD10CM/N18.30 within 1095d | no TIMELINE//END within 1095d	0.090	**0.895**	0.721
ICD10CM/N18.3 within 1095d | no TIMELINE//END within 1095d	0.073	**0.930**	0.763
ICD10CM/I50.9 within 1095d | no TIMELINE//END within 1095d	0.071	**0.925**	0.774
ICD10CM/I50.22 within 1095d | no TIMELINE//END within 1095d	0.070	**0.881**	0.742
RxNorm/617310 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.064	**0.709**	0.694
RxNorm/617320 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.050	**0.800**	0.755
RxNorm/617311 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.050	**0.794**	0.731
RxNorm/312961 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.044	**0.761**	0.755
ICD10CM/I50.20 within 1095d | no TIMELINE//END within 1095d	0.043	**0.896**	0.693
RxNorm/259255 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.042	**0.847**	0.742
ICD10CM/I50.32 within 1095d | no TIMELINE//END within 1095d	0.041	**0.853**	0.656

continued on next page

Task	Prev.	EveryQuery	Autoregressive
RxNorm/198211 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.039	**0.797**	0.778
ICD10CM/I50.23 within 1095d | no TIMELINE//END within 1095d	0.034	**0.884**	0.636
ICD10CM/I50.30 within 1095d | no TIMELINE//END within 1095d	0.034	**0.842**	0.618
RxNorm/617314 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.031	**0.716**	0.704
RxNorm/262095 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.026	**0.842**	0.772
RxNorm/617318 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.026	**0.743**	0.722
RxNorm/617312 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.026	0.711	**0.744**
ICD10CM/I50.810 within 1095d | no TIMELINE//END within 1095d	0.024	**0.869**	0.636
ICD10CM/E11.3593 within 1095d | no TIMELINE//END within 1095d	0.021	**0.907**	0.735
ICD10CM/I50.33 within 1095d | no TIMELINE//END within 1095d	0.020	**0.886**	0.564
RxNorm/904475 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.020	**0.812**	0.626
ICD10CM/I50.21 within 1095d | no TIMELINE//END within 1095d	0.019	**0.837**	0.575
RxNorm/904467 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.016	**0.863**	0.618
RxNorm/314231 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.015	**0.732**	0.690
RxNorm/859753 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.014	**0.768**	0.724
RxNorm/859749 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.014	**0.710**	0.682
CPT4/92250 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.014	**0.773**	0.634
ICD10CM/I50.31 within 1095d | no TIMELINE//END within 1095d	0.013	**0.854**	0.572
ICD10CM/N18.31 within 1095d | no TIMELINE//END within 1095d	0.013	**0.959**	0.653
RxNorm/104491 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.011	**0.835**	0.719
RxNorm/859421 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.011	**0.874**	0.800

continued on next page

Task	Prev.	EveryQuery	Autoregressive
RxNorm/152923 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.010	**0.853**	0.793
ICD10CM/I50.43 within 1095d | no TIMELINE//END within 1095d	0.010	**0.896**	0.545
ICD10CM/N18.32 within 1095d | no TIMELINE//END within 1095d	0.010	**0.965**	0.688
ICD10CM/I50.1 within 1095d | no TIMELINE// /END within 1095d	0.009	**0.930**	0.600
RxNorm/859747 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.009	**0.866**	0.695
ICD10CM/I50.42 within 1095d | no TIMELINE//END within 1095d	0.009	**0.873**	0.602
RxNorm/859751 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.009	**0.823**	0.705
RxNorm/897126 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.009	**0.860**	0.735
RxNorm/859419 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.007	**0.860**	0.816
RxNorm/200345 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.007	**0.832**	0.768
RxNorm/1545664 within 365d | no TIMELINE// /END within 365d, no MEDS_DEATH within 365d	0.007	**0.932**	0.770
RxNorm/859426 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.007	**0.758**	0.697
ICD10CM/E11.621 within 730d | no TIMELINE//END within 730d	0.007	**0.928**	0.656
RxNorm/904458 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.006	**0.796**	0.661
ICD10CM/E11.649 within 365d | no TIMELINE//END within 365d, no MEDS_DEATH within 365d	0.006	**0.926**	0.547
ICD10CM/E11.3293 within 1095d | no TIMELINE//END within 1095d	0.006	**0.859**	0.579
RxNorm/859424 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.006	**0.921**	0.660
RxNorm/904477 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.006	**0.818**	0.679
RxNorm/904469 within 365d | no TIMELINE// END within 365d, no MEDS_DEATH within 365d	0.005	**0.875**	0.546
RxNorm/1551300 within 365d | no TIMELINE// /END within 365d, no MEDS_DEATH within 365d	0.004	**0.912**	0.702

Table 7: MIMIC-IV ICU, 24h after admission ($n = 38$ tasks).

Task	Prev.	AUROC	
		EveryQuery	Autoregressive
ICU_DISCHARGE within 7d | no TIMELINE// END within 7d, no MEDS_DEATH before ICU_DISCHARGE	0.882	**0.782**	0.710
ICU_DISCHARGE within 3d | no TIMELINE// END within 3d, no MEDS_DEATH before ICU_DISCHARGE	0.712	**0.796**	0.629
ICU_DISCHARGE within 2d | no TIMELINE// END within 2d, no MEDS_DEATH before ICU_DISCHARGE	0.599	**0.798**	0.595
ICU_DISCHARGE within 1d | no TIMELINE// END within 1d, no MEDS_DEATH before ICU_DISCHARGE	0.390	**0.798**	0.564
MEDS_DEATH within 365d	0.293	**0.831**	0.727
MEDS_DEATH within 90d	0.207	**0.849**	0.747
LAB//RESULT//50983//mEq/L// value_[-inf,133.0) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.150	**0.743**	0.668
MEDS_DEATH within 28d	0.146	**0.874**	0.771
LAB//RESULT//50912//mg/dL// value_[2.3,inf) within 3d | no TIMELINE// END within 3d, no MEDS_DEATH within 3d	0.143	**0.899**	0.789
MEDICATION//START//Vancomycin within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.135	**0.749**	0.662
LAB//RESULT//50912//mg/dL// value_[2.3,inf) within 2d | no TIMELINE// END within 2d, no MEDS_DEATH within 2d	0.134	**0.903**	0.797
LAB//RESULT//50882//mEq/L// value_[-inf,20.0) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.134	**0.816**	0.755
LAB//RESULT//51222//g/dL// value_[-inf,7.9) within 1d | no TIMELINE/ /END within 1d, no MEDS_DEATH within 1d	0.132	**0.801**	0.727
LAB//RESULT//50971//mEq/L// value_[-inf,3.5) within 1d | no TIMELINE/ /END within 1d, no MEDS_DEATH within 1d	0.125	**0.711**	0.658
LAB//RESULT//51265//K/uL// value_[-inf,86.0) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.123	**0.849**	0.737
LAB//RESULT//50931//mg/dL// value_[191.0,inf) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.116	**0.823**	0.732
MEDS_DEATH before ICU_DISCHARGE	0.072	**0.860**	0.810
LAB//RESULT//50885//mg/dL// value_[2.6,inf) within 3d | no TIMELINE// END within 3d, no MEDS_DEATH within 3d	0.072	**0.904**	0.739

continued on next page

Task	Prev.	EveryQuery	Autoregressive
LAB//RESULT//50821//mm Hg// value_[-inf,43.0) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.056	**0.859**	0.711
LAB//RESULT//51237//UNK//value_[2.6,inf) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.053	**0.790**	0.644
MEDICATION//START//CefePIME within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.043	**0.807**	0.650
MEDS_DEATH within 3d	0.040	**0.939**	0.833
LAB//RESULT//50820//units// value_[-inf,7.26) within 0.5d | no TIMELINE//END within 0.5d, no MEDS_DEATH within 0.5d	0.024	**0.904**	0.831
MEDICATION//START// Piperacillin-Tazobactam within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.024	**0.833**	0.648
PROCEDURE//START//225792 within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.019	**0.785**	0.499
LAB//RESULT//50813//mmol/L// value_[4.0,inf) within 0.5d | no TIMELINE//END within 0.5d, no MEDS_DEATH within 0.5d	0.019	**0.923**	0.898
MEDS_DEATH within 1d	0.016	**0.959**	0.894
LAB//RESULT//50813//mmol/L// value_[4.0,inf) within 0.25d | no TIMELINE//END within 0.25d, no MEDS_DEATH within 0.25d	0.015	**0.945**	0.938
INFUSION_START//221906// value_[0.32051384,inf) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.015	**0.952**	0.763
INFUSION_START//221906// value_[0.32051384,inf) within 0.5d | no TIMELINE//END within 0.5d, no MEDS_DEATH within 0.5d	0.013	**0.958**	0.924
MEDICATION//START//Levofloxacin within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.012	**0.840**	0.604
PROCEDURE//START//225441 within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.011	**0.931**	0.600
PROCEDURE//START//224385 within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.011	**0.813**	0.498
PROCEDURE//START//225792 within 0.5d | no TIMELINE//END within 0.5d, no MEDS_DEATH within 0.5d	0.011	**0.814**	0.638
MEDICATION//START//Ciprofloxacin HCl within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.010	**0.776**	0.522

continued on next page

Task	Prev.	EveryQuery	Autoregressive
MEDICATION//START//Meropenem within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.008	**0.897**	0.590
INFUSION_START//221749//value_[3.0013728,inf) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.007	**0.919**	0.755
PROCEDURE//START//225794 within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.006	**0.829**	0.555

Table 8: NWICU ICU, 24h after admission ($n = 27$ tasks).

Task	Prev.	AUROC	
		EveryQuery	Autoregressive
ICU_DISCHARGE within 7d | no TIMELINE//END within 7d, no MEDS_DEATH before ICU_DISCHARGE	0.875	**0.721**	0.528
ICU_DISCHARGE within 3d | no TIMELINE//END within 3d, no MEDS_DEATH before ICU_DISCHARGE	0.686	**0.706**	0.619
ICU_DISCHARGE within 2d | no TIMELINE//END within 2d, no MEDS_DEATH before ICU_DISCHARGE	0.565	**0.688**	0.661
ICU_DISCHARGE within 1d | no TIMELINE//END within 1d, no MEDS_DEATH before ICU_DISCHARGE	0.369	0.676	**0.693**
MEDS_DEATH within 365d	0.203	**0.817**	0.737
MEDS_DEATH within 90d	0.162	**0.850**	0.781
MEDS_DEATH within 28d	0.133	**0.874**	0.809
PROCEDURE//START//787541 within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.131	0.955	**0.963**
PROCEDURE//START//792843 within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.109	0.823	**0.908**
LAB//100001//mg/dL//value_[236.0,inf) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.102	**0.899**	0.888
LAB//100002//mg/dL//value_[3.27,inf) within 3d | no TIMELINE//END within 3d, no MEDS_DEATH within 3d	0.094	**0.957**	0.956
PROCEDURE//START//787541 within 0.5d | no TIMELINE//END within 0.5d, no MEDS_DEATH within 0.5d	0.091	0.935	**0.956**
LAB//100002//mg/dL//value_[3.27,inf) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.089	**0.963**	0.962
PROCEDURE//START//792843 within 0.5d | no TIMELINE//END within 0.5d, no MEDS_DEATH within 0.5d	0.075	0.787	**0.898**

continued on next page

Task	Prev.	EveryQuery	Autoregressive
LAB//100007//g/dL//value_[-inf,7.4) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.070	**0.920**	0.918
MEDS_DEATH before HOSPITAL_DISCHARGE	0.070	**0.839**	0.783
LAB//100013//mEq/L//value_[-inf,20.0) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.064	0.900	**0.933**
LAB//100011//mEq/L//value_[-inf,3.4) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.062	0.841	**0.860**
LAB//100010//mEq/L//value_[-inf,132.0) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.047	0.835	**0.858**
LAB//100020//mg/dL//value_[3.0,inf) within 3d | no TIMELINE//END within 3d, no MEDS_DEATH within 3d	0.044	**0.963**	0.935
MEDS_DEATH before ICU_DISCHARGE	0.044	**0.842**	0.782
LAB//100011//mmol/L//value_[-inf,3.4) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.042	**0.893**	0.870
MEDS_DEATH within 3d	0.038	**0.933**	0.886
LAB//100013//mmol/L//value_[-inf,20.0) within 1d | no TIMELINE//END within 1d, no MEDS_DEATH within 1d	0.032	0.926	**0.932**
LAB//100014//10^3/μL//value_[-inf,82.0) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.031	**0.961**	0.959
LAB//100010//mmol/L//value_[-inf,132.0) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.029	**0.915**	0.912
LAB//100014//K/UL//value_[-inf,56.0) within 2d | no TIMELINE//END within 2d, no MEDS_DEATH within 2d	0.026	**0.969**	0.957

D Results on Time-to-Event Tasks

D.1 Target codes and prediction horizons

Table 9 lists the target codes used for time-to-event evaluation in each dataset. For every code, we evaluate the survival curve at 16 horizons: 1, 2, 3, 5, 7, 10, 14, 21, 30, 45, 60, 90, 120, 180, 270 and 365 days.

Table 9: **Target codes for time-to-event evaluation.** Codes are written as they appear in each dataset's vocabulary.

Dataset	Target code

MIMIC-IV	`MEDS_DEATH`
	`PROCEDURE//START//225792`
	`ICU_ADMISSION//Medical Intensive Care Unit (MICU)`
	`MEDICATION//START//Norepinephrine`
	`HOSPITAL_DISCHARGE//HOME`
	`INFUSION_START//221289//value_[0.35491893,inf)`
	`INFUSION_START//221749//value_[3.0013728,inf)`

NWICU	`DEATH`
	`PROCEDURE//START//787541`
	`PROCEDURE//START//792843`
	`DIAGNOSIS//ICD//10//A419`
	`HOSPITAL_DISCHARGE`

AMC	`MEDS_DEATH`
	`ICD10CM/I48.0`
	`Visit/IP`
	`Visit/ER`
	`LOINC/2160-0//UCUM/mg/dL//value_[2.34,inf)`
	`LOINC/2823-3//UCUM/mmol/L//value_[5.0,inf)`
	`ICD10CM/N18.3`
	`RxNorm/1373469`
	`RxNorm/198222`

D.2 SURVIVAL CURVES

Figures 4, 5 and 6 compare the survival curves estimated by EveryQuery and the autoregressive baseline with the Kaplan–Meier estimate for every target code on MIMIC-IV, NWICU and AMC.

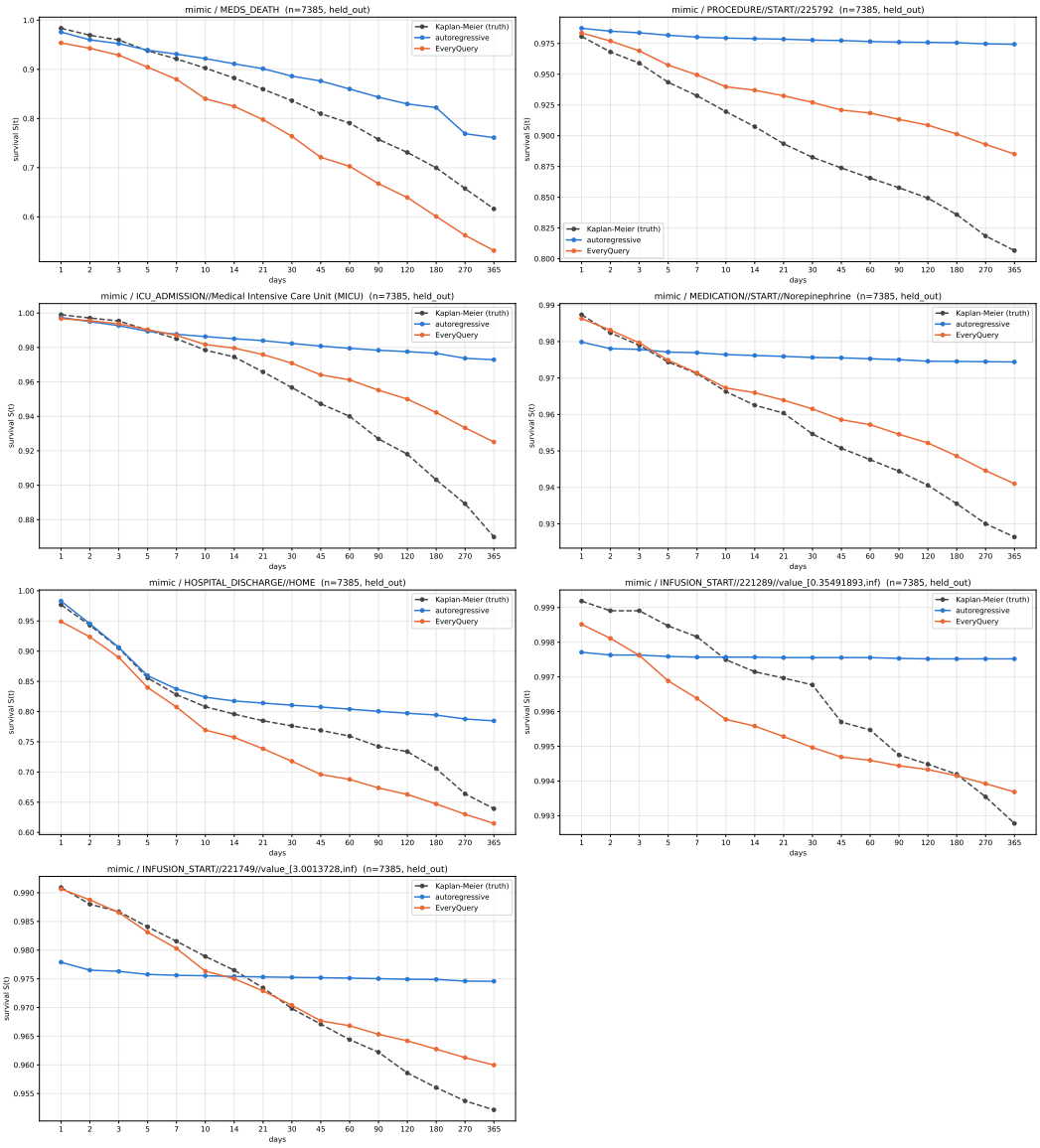

Figure 4: **Survival curves on MIMIC-IV.** Each panel shows one target code (Table 9), comparing the Kaplan–Meier estimate on held-out patients (black, dashed) with the survival curves estimated by the autoregressive baseline (blue) and EveryQuery (orange) at each prediction horizon in days.

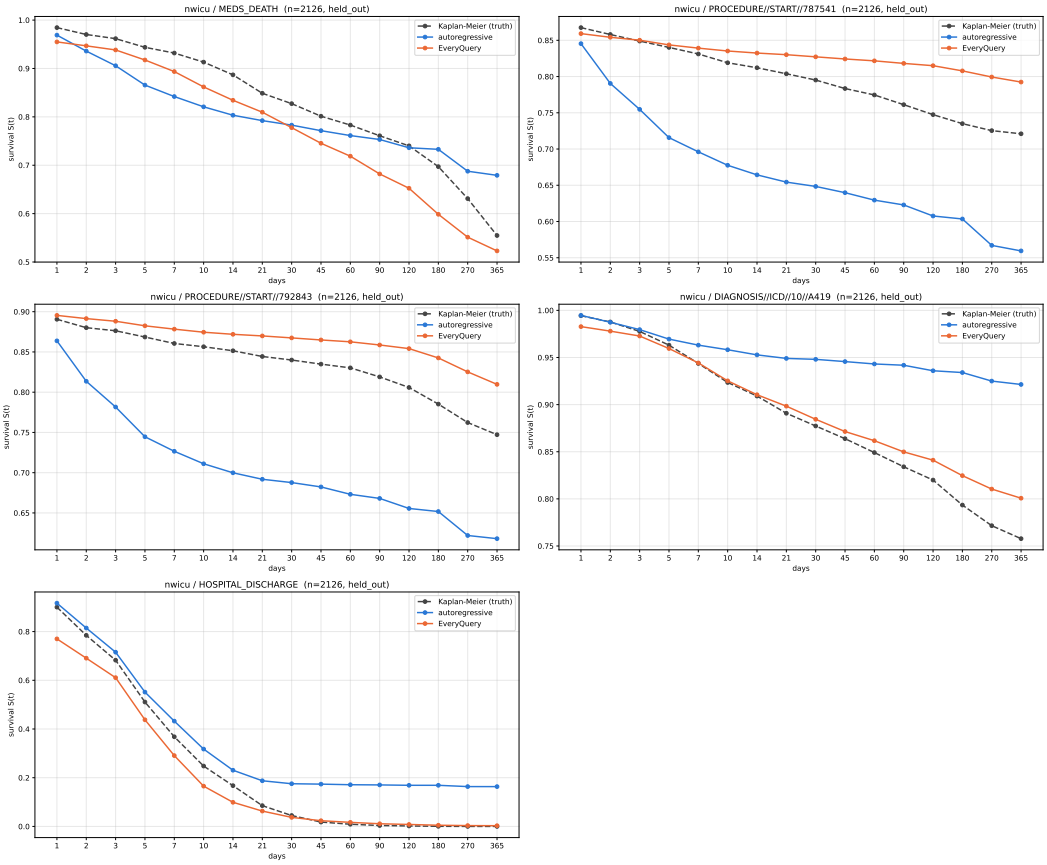

Figure 5: **Survival curves on NWICU.** Each panel shows one target code (Table 9), comparing the Kaplan–Meier estimate on held-out patients (black, dashed) with the survival curves estimated by the autoregressive baseline (blue) and EveryQuery (orange) at each prediction horizon in days.

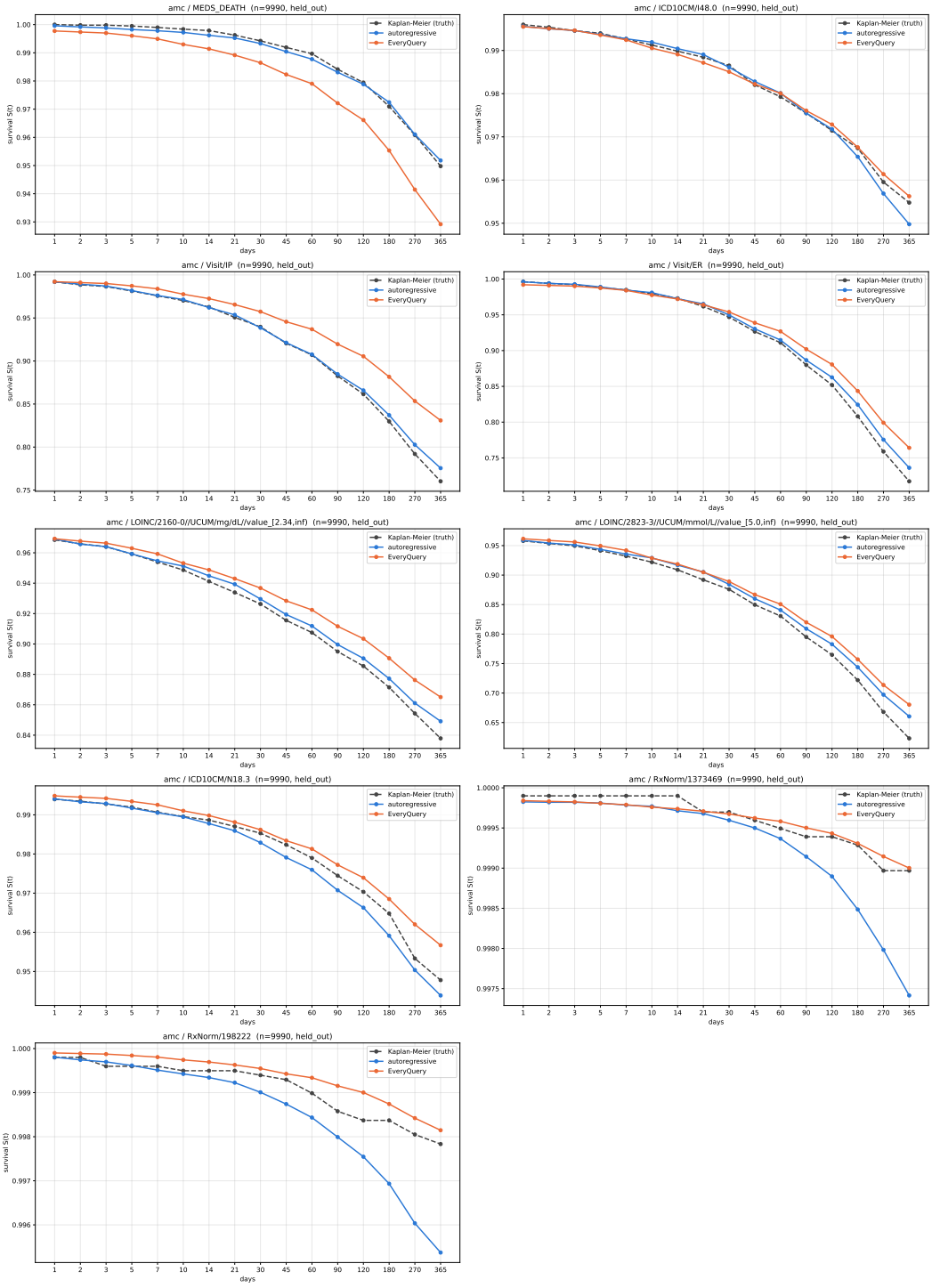

Figure 6: **Survival curves on AMC.** Each panel shows one target code (Table 9), comparing the Kaplan–Meier estimate on held-out patients (black, dashed) with the survival curves estimated by the autoregressive baseline (blue) and EveryQuery (orange) at each prediction horizon in days.

E Results on Randomly Sampled Tasks

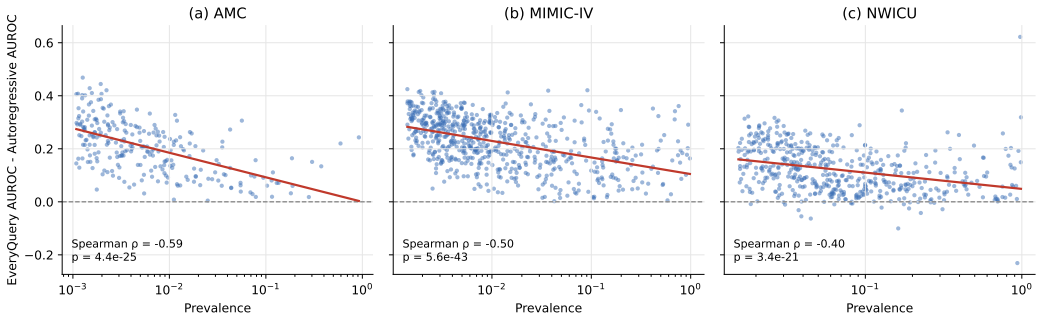

Figure 7: **On randomly sampled tasks, EveryQuery's advantage over the autoregressive baseline grows as events become rarer.** Each point is one randomly sampled task (§5.5), plotted by prevalence (log scale) against ΔAUROC (EveryQuery minus autoregressive). Red lines show linear fits, and each panel reports the Spearman correlation between prevalence and ΔAUROC.