EhrMamba: Towards Generalizable and Scalable Foundation Models for Electronic Health Records

Adibvafa Fallahpour[1,2*]   ADIBVAFA.FALLAHPOUR@VECTORINSTITUTE.AI

Mahshid Alinoori[1*]   MAHSHID.ALINOORI@VECTORINSTITUTE.AI

Wenqian Ye[3]   WENQIAN@VIRGINIA.EDU

Xu Cao[4]   XUCAO2@ILLINOIS.EDU

Arash Afkanpour[1†]   ARASH.AFKANPOUR@VECTORINSTITUTE.AI

Amrit Krishnan[1†]   AMRIT.KRISHNAN@VECTORINSTITUTE.AI

[1]*Vector Institute, Canada*   [2]*University of Toronto, Canada*   [3]*University of Virginia, USA*
[4]*University of Illinois Urbana-Champaign, USA*

Abstract

Transformers have significantly advanced the modeling of Electronic Health Records (EHR), yet their deployment in real-world healthcare is limited by several key challenges. Firstly, the quadratic computational cost and insufficient context length of these models hinder hospitals' ability in processing the extensive medical histories typical in EHR data. Additionally, existing models employ separate finetuning for each clinical task, complicating maintenance in healthcare environments. Moreover, these models focus exclusively on either clinical prediction or EHR forecasting, lacking proficiency in both tasks. To overcome these limitations, we introduce **EhrMamba**, a robust foundation model built on the Mamba architecture. **EhrMamba** can process sequences up to 300% longer than previous models due to its linear computational cost. We also introduce a novel approach to Multi-task Prompted Finetuning (MPF) for EHR data, which enables **EhrMamba** to simultaneously learn multiple clinical tasks in a single finetuning phase, significantly enhancing deployment and cross-task generalization. Furthermore, our model leverages the HL7 FHIR data standard to simplify integration into existing hospital systems. Alongside **EhrMamba**, we open-source **Odyssey**, a toolkit designed to support the development and deployment of EHR foundation models, with an emphasis on data standardization and interpretability. Our evaluations on the MIMIC-IV dataset demonstrate that **EhrMamba** advances state-of-the-art performance across 6 major clinical tasks and excels in EHR forecasting, marking a significant leap forward in the field.

Keywords: Mamba, Electronic Health Records, Foundation Model, Healthcare, Odyssey

Data and Code Availability The MIMIC-IV dataset is publicly available here (20; 21; 32; 16). All models were developed and evaluated in this study using the **Odyssey** toolkit.

1. Introduction

Personalized medicine is the pinnacle of healthcare innovation, and AI represents a promising solution (9). Central to this revolution are Electronic Health Records (EHR), which document the entire medical histories of patients in hospital visits (56). These records form detailed chronological sequences that include diagnoses, procedures, observations, medications, laboratory tests, demographics, and clinical notes of millions of patients over decades. With over 80% of hospitals in the US and Canada adopting EHR systems, this extensive data provides an unparalleled resource for training EHR foundation models (1; 44). These models hold the potential to personalize treatment plans, uncover disease patterns, detect the onset of rare illnesses, and enhance clinical predictions (56; 40; 27).

Transformer-based models, particularly variants of BERT (10), have demonstrated remarkable capabil-

---

* Equal contribution
† Equal advising

ities in modeling EHR data and predicting clinical outcomes (38; 27; 44; 41; 28; 26). However, their translation to real-world clinical settings remains an open challenge. Existing models often prioritize research performance and may not adequately consider the practical constraints faced by hospitals for deploying such large-scale models (40; 56). These include limited computational resources, data privacy regulations mandating on-premise deployments, and the need for flexible models that generalize to new tasks and integrate seamlessly with existing healthcare infrastructure. In this paper, we mainly focus on the following bottlenecks:

Computational Constraints. Deploying Transformer-based models in hospital settings is challenged by the length of EHR data. The quadratic scaling of computational and memory requirements becomes prohibitive when sequences span thousands of tokens to capture a patient's entire medical history. Early diagnoses, noted at the beginning of these sequences, influence the interpretation of visits occurring decades later, requiring extensive context lengths to effectively capture these longitudinal healthcare dynamics. However, the computational resources required often far exceed what is available in many hospitals.

Finetuning Overhead. Finetuning EHR models for each downstream clinical predictive task, such as predicting patient mortality, presents several significant challenges. This process entails creating a separate copy of the base pretrained model and finetuning it for each specific task, leading to the simultaneous management and maintenance of multiple specialized models. Finetuning task-specific models in isolation demands considerable resources and hinders their ability to integrate insights across different tasks, impairing model's reliability and performance.

To overcome these limitations, we propose **EhrMamba**, a robust foundation model designed for scalable, deployable, and generalizable EHR modeling. **EhrMamba** is built on the Mamba architecture and introduces several key contributions:

Scalability. **EhrMamba** extends the context length by 300% compared to previous transformer-based models, enabling the processing of longer EHR sequences and the inclusion of more comprehensive information in the sequence (56).

Multitask Prompted Finetuning (MPF). We train **EhrMamba** using a variant of MPF (53) for EHR data, allowing simultaneous learning of multiple clinical predictive tasks within a single finetuning phase. This approach enhances cross-task generalization, supports the learning of new tasks without modifying the model architecture, and simplifies real-world deployment in hospitals.

Dual Competence. **EhrMamba** is the first model to perform both EHR forecasting, predicting future data in EHR sequences, and clinical predictive modeling, predicting patient outcomes such as mortality. This dual functionality enables comprehensive disease pattern forecasting and personalized prediction timelines, facilitating tailored treatment plans based on individual patient trajectories.

Odyssey. **EhrMamba** is built using **Odyssey**, a toolkit designed to facilitate the development and deployment of EHR foundation models. **Odyssey** supports gathering and processing EHR sequences using the HL7 Fast Healthcare Interoperability Resources (FHIR) standard, which simplifies integration into existing hospital systems due to its widespread adoption in healthcare settings (32). **Odyssey** comprises four key modules: data processing, model implementations, evaluation tools, and interpretation methods, providing a comprehensive framework for EHR-based machine learning research. See Appx.A. for more information on **Odyssey**.

We assess **EhrMamba** on 6 major clinical predictive tasks using the MIMIC-IV dataset (20; 21; 16). Our results show that **EhrMamba** achieves state-of-the-art performance while operating with significant memory and computational efficiency. Additionally, we present a patient case study highlighting EHR forecasting capabilities and interpretability methods.

2. Preliminaries

2.1. Data Representation.

Patient sequences in EHR data represent a time series of medical events encoded using standard biomedical tokens (39; 48; 47; 6). Each patient sequence starts with a [CLS] token, similar to the classification token used in BERT (10), followed by a series of visits marked by [VS] (visit start) and [VE] (visit end) tokens. Special tokens indicating the time interval are used between consecutive visits, e.g. $[W_2]$ represents

		Visit 1							Visit 2							
Concepts	CLS	VS	P1	M1	M2	VE	REG	W₂	VS	L1	L2	L3	P2	VE	REG	PAD
Token Types	0	1	2	3	3	4	5	6	2	7	7	7	2	4	5	8
Age	0	46	46	46	46	46	46	0	47	47	47	47	47	47	47	0
Time	0	53	53	53	53	53	53	0	55	55	55	55	55	55	55	0
Segment	0	1	1	1	1	1	1	0	2	2	2	2	2	2	2	0
Visit Order	0	1	1	1	1	1	1	0	2	2	2	2	2	2	2	0
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	l_c-1
Embedded Model Input	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}	E_{13}	E_{14}	0

Figure 1: Patient sequence example. Visit 1 has a procedure and two medications; visit 2, after two weeks, has three lab tests and another procedure. The concept embedding of each token is added to its attribute embeddings (type, age, time, segment, visit order, position) to encode the sequence.

2 weeks (38), and each visit end token is followed by a register token [REG], similar to its use in Vision Transformers (8; 11). Appx.E. provides more information.

Medical events within visits are organized chronologically and represented by tokens $e_{v,j}$, where v denotes the visit number, and j is the index of the event within that visit (38). Each token $e_{v,j}$ is associated with a timestamp $\tau_{v,j}$, indicating the exact time of the event such that $\tau_{v,j} \leq \tau_{v,j+1}$. We define a function $f(e_{v,j}) : \mathcal{E} \rightarrow \Gamma$ that maps the event space $\mathcal{E}$ to a set of attributes Γ including: (57):

- T: Type of the event, either procedure (P), medication (M), or lab result (L), see Appx.B.
- A: Age of the patient at the time of the event.
- τ: Exact timestamp of the event, counted in weeks relative to some reference.
- S: Visit segment, alternating between 1 and 2 to indicate the separation between adjacent visits.
- V: Visit order, the absolute visit number incrementing for each subsequent visit.
- $\mathcal{P}$: Absolute position of the event.

Each attribute is mapped to a distinct token space, and their embeddings are added to the embeddings of event tokens, referred to as the concept embeddings. The sequence is padded to context length l_c. Figure 1 shows an example patient representation and the corresponding embeddings.

2.2. Embeddings.

The embedding of event token $e_{v,j}$, denoted as $E_{v,j}$, is the sum of its concept embedding with the respective attribute embeddings:

$$\begin{aligned} E_{v,j} = \ &E_{\text{concept}}(e_{v,j}) + E_{\text{type}}(T_{v,j}) + E_{\text{age}}(A_{v,j}) \\ &+ E_{\text{time}}(\tau_{v,j}) + E_{\text{segment}}(S_{v,j}) \\ &+ E_{\text{visit order}}(V_{v,j}) + E_{\text{position}}(\mathcal{P}_{v,j}) \end{aligned} \quad (1)$$

Here, $E_{v,j} \in \mathbb{R}^d$, where d is embedding size (27; 38).

The embeddings are learned as follows:

- **Concept Embeddings**: Each token $e_{v,j}$ maps to an embedding, like word embeddings (15; 34).
- **Token Type Embeddings**: Embeddings representing the type of medical event or special token.
- **Age Embeddings**: Continuous embeddings denoting age at the time of event, see Appx.C.2 (38).
- **Time Embeddings**: Continuous embeddings for elapsed time since reference, see Appx.C.2 (38).
- **Visit Segment Embeddings**: Embeddings indicating segment 1 or 2.
- **Visit Order Embeddings**: Embeddings representing the sequential order of visits.
- **Positional Embeddings**: Absolute positional embeddings (10; 61).

Special tokens have their own concept, token type, and positional embeddings, while for other embeddings, a zero vector is used (27). This effectively captures the rich temporal and contextual information necessary for modeling patient sequences (47; 48; 58; 4).

2.3. Learning Objectives.

The overarching goal in EHR predictive modeling is to optimize predictions of patient outcomes, such as predicting patient mortality, given patient sequences. However, given the scarcity of labeled data in healthcare settings, a two-phase learning approach is employed: First, pretraining leverages the abundant unlabeled EHR data to capture temporal dynamics and general patterns in patient sequences. Subsequently, finetuning over a small amount of labeled data adapts this prelearned knowledge to specific clinical tasks.

2.3.1. PRETRAINING.

Pretraining can be approached in two primary ways, depending on the model type: Next Token Prediction (NTP) and Masked Language Modeling (MLM).

Next Token Prediction (NTP). In NTP, the model predicts the next event token $e_{v,j+1}$ in a patient sequence, given all preceding tokens in the sequence, denoted as $\mathcal{I} = \{e_{1:v,1:j}\}$:

$$\theta^*_{\text{NTP}} = \arg\min_{\theta} \mathbb{E}_{\mathcal{D}} \left[\sum_{(v,j) \in \mathcal{I}} -\log p(e_{v,j+1}|e_{1:v,1:j};\theta) \right] \tag{2}$$

Here, $\mathbb{E}_{\mathcal{D}}$ denotes the expectation over the empirical distribution of the dataset $\mathcal{D}$. Optimized model parameters are denoted by θ^*.

Masked Language Modeling (MLM). In MLM, the model predicts the identity of masked event tokens, $e_{v,j} \in \mathcal{M}$, given the unmasked tokens $e_{\mathcal{C}(v,j)}$ as context. The set $\mathcal{C}(v,j)$ is defined as all tokens in the sequence except the masked tokens.

$$\theta^*_{\text{MLM}} = \arg\min_{\theta} \mathbb{E}_{\mathcal{D}} \left[\sum_{(v,j) \in \mathcal{M}} -\log p(e_{v,j}|e_{\mathcal{C}(v,j)};\theta) \right] \tag{3}$$

2.3.2. FINE-TUNING.

In clinical predictive modeling, each task c_i from a set of $\{c_i\}_{i=1}^{K}$ tasks is treated as a binary classification problem with outcomes $y_{c_i} \in \{0, 1\}$. The traditional method finetunes model parameters θ_{c_i} separately for each task, expressed as:

$$\{\theta^*_{c_i}\}_{i=1}^{K} = \left\{ \arg\min_{\theta_{c_i}} \mathcal{L}_i(\hat{y}_{c_i}(x;\theta_{c_i}), y_{c_i}) \right\}_{i=1}^{K} \tag{4}$$

where $\mathcal{L}_i$ is the binary cross-entropy loss and $\hat{y}_{c_i}(x;\theta_{c_i})$ is the predicted outcome for task c_i. This approach isolates the optimization for each task, preventing the model from leveraging shared knowledge across tasks. Such isolation complicates deployment and maintenance in healthcare, as each task-specific model requires separate management, increasing overhead.

3. EHRMAMBA

3.1. State Space Models (SSMs)

The Mamba architecture is a type of State Space Model (SSM), a powerful class of sequence models that have proven effective at handling long-range sequences (17; 18). SSMs map an input sequence $x(t) \in \mathbb{R}$ to an output sequence $y(t) \in \mathbb{R}$ via an implicit hidden state $h(t) \in \mathbb{R}^N$, where N is the state size. The mapping is defined by the following linear differential equations:

$$h'(t) = Ah(t) + Bx(t), \qquad y(t) = Ch(t) \tag{5}$$

Here, $A \in \mathbb{R}^{N \times N}$, $B \in \mathbb{R}^{N \times 1}$, and $C \in \mathbb{R}^{1 \times N}$ are learnable matrices. For multidimensional sequences, this system is applied independently to each dimension (17; 18; 45).

3.1.1. DISCRETIZATION OF THE SYSTEM

To apply SSMs to discrete sequences, the system is discretized using a step size Δ: (17; 59)

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t, \qquad y_t = Ch_t \tag{6}$$

The discrete parameters $(\overline{A}, \overline{B})$ are derived using the zero-order hold (ZOH) rule: (17; 18)

$$\overline{A} = \exp(\Delta A) \qquad \overline{B} = (\Delta A)^{-1} \left((\exp(\Delta A) - I) \cdot \Delta B\right) \tag{7}$$

3.1.2. EXECUTION METHODS

Linear Recurrence for Inference. The model is computed step-by-step as a linear recurrence relation, as shown in Equation (6). This results in fast inference and linear scaling in relation to sequence length, enhancing deployability in resource-limited hospitals.

Global Convolution for Training. The model uses a global convolution kernel $\overline{K}$ to enable parallel sequence processing and scalable training, described by Equation (9) below: (17; 31; 59; 54)

$$K = (C\overline{B}, C\overline{A}\overline{B}, \ldots, C\overline{A}^k\overline{B}, \ldots) \qquad y = x * \overline{K} \tag{8}$$
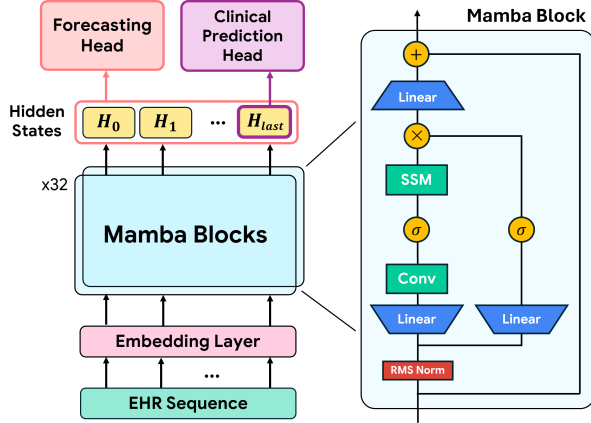
4

Figure 2: **EHRMAMBA** architecture. Pretraining uses the forecasting head and finetuning uses the clinical prediction head.

3.2. EHRMamba Architecture

EHRMamba comprises three main components: a specialized embedding layer for EHR training, multiple Mamba blocks, and two custom heads. One is used for forecasting—predicting subsequent events and visits in the patient sequence—and the other for clinical predictions, such as predicting patient mortality. Figure 2 provides an overview of the EHRMamba architecture.

3.2.1. EMBEDDING LAYER

The embedding layer maps input sequences $\mathbf{X} \in \mathbb{R}^{b \times l_c}$ to embedded input $\mathbf{E} \in \mathbb{R}^{b \times l_c \times d}$ where b is the batch size. Details of the embedding layer in EHRMamba are provided in Section 2.2. Given that Mamba inherently handles sequential data, positional embeddings are excluded (17; 19).

3.2.2. MAMBA BLOCKS

The core of EHRMamba is composed of stacked Mamba blocks, each acting as a sequence-to-sequence module that preserves the input and output dimensions. These blocks map the input embeddings to the output tensor $H \in \mathbb{R}^{b \times l_c \times d}$. Refer to Appx.C.1 for the implementation of Mamba blocks.

3.2.3. FORECASTING HEAD - PRETRAINING

This is the causal language modeling head that maps the last hidden states onto the vocabulary. The input first passes through a RMS normalization (62),

followed by a linear projection and softmax:

$$\hat{\mathbf{Y}} = Softmax(Linear(RMSNorm(\mathbf{H}))) \qquad (9)$$

Here, $\mathbf{H}$ and $\hat{\mathbf{Y}} \in \mathbb{R}^{b \times l_c \times v}$ represent the last hidden states and forecasting outputs, respectively, where v is the vocabulary size.

3.2.4. CLINICAL PREDICTION HEAD - FINETUNING

This is a classification head that maps $\mathbf{H}_{\text{last}}$, the last hidden state of the last token before padding in the patient sequence, to a predicted probability $\hat{y}_c$ for a given clinical prediction task (46; 12):

$$\hat{\mathbf{y}}_{\mathbf{c}} = Sigmoid(Linear(Dropout(\mathbf{H}_{\text{last}}))) \qquad (10)$$

The Multitask Prompted Finetuning (MPF) (53; 35) approach discussed below enables the same trained head to be used for predicting multiple clinical tasks.

3.3. Multitask Prompted Finetuning (MPF)

The goal of Multitask Prompted Finetuning (MPF) is to enable a single finetuned model to excel across multiple clinical tasks (53; 35). MPF involves a preprocessing step that replaces the first and last tokens of the sequence, a [CLS] and [REG] token respectively, with special task-specific tokens, such as [MOR] for mortality prediction. Predictions are then derived from the last hidden state of the task-specific token. This allows the same patient sequence data to be used for various clinical tasks, each distinguished by its specific task token. During finetuning, patient sequences with task-specific tokens and appropriate labels are processed concurrently.

MPF brings the following advantages:

- **Deployment.** MPF streamlines finetuning across multiple tasks and facilitates deployment on new tasks by simply introducing task-specific tokens. While using separate classification heads per task is possible, maintaining these isolated heads becomes challenging as the number of tasks grows.

- **Learning.** MPF embeds task-specific information at the input level, enhancing the model's ability to tailor its processing and final hidden states to the specific task. This leads to efficient learning and enhances generalization across tasks, as the model is aware of the task context from the beginning.

- **Compatibility.** MPF utilizes the identical model architectures used in popular frameworks like HuggingFace (55). This compatibility facilitates the seamless transfer and reuse of trained weights, simplifying integration with existing systems and reducing deployment overhead (35).

4. Experiments

4.1. Experimental Setup

Dataset. We evaluate **EHRMAMBA** on MIMIC-IV, a real-world, publicly available EHR dataset from Beth Israel Deaconess Medical Center (20; 21; 16). It includes records from over 431,000 visits and 180,000 patients, featuring detailed temporal information on medical events such as procedures, medications, and lab results, along with demographic information such as age. Refer to Appx.B. for more information on the dataset and data splits.

Clinical Predictive Tasks. We assess models' performance on 6 primary clinical predictive binary classification tasks: (1) Mortality Prediction, predicting whether a patient will pass away within one month after hospital discharge (50; 13; 23), (2) Length of Stay Prediction, estimating whether a patient's hospitalization will exceed one week based on the first 24 hours of admission (29; 40), (3) Readmission Prediction, predicting the likelihood of a patient being readmitted within one month of the most recent discharge (37; 52; 7), along with predicting for three specific diagnostic conditions: (4) Condition 0 (Hypertension), (5) Condition 1 (Fluid Disorders), (6) Condition 2 (Lipoid Metabolism Disorders) (2; 43; 42). Refer to Appx.B. for more information.

Evaluation Metrics. We evaluate model performances using the Area Under the Receiver Operating Characteristic Curve (AUROC), the Area Under the Precision-Recall Curve (AUPRC), and the F1-Score metrics (33; 27; 41; 44). We calculate averages and standard deviations by conducting experiments three times with randomized seeds, and use independent two-sample T-tests to assess the statistical significance of our results (33).

4.2. Baseline Models.

We compare **EHRMAMBA** to 5 baseline models described below. All these models, except XGBoost, use the embedding layer describe in Section 2.2. Additionally, except MultiBird and EHRMamba, other models are trained or finetuned separately for each clinical task. Implementation and training details are provided in Appx.C.

XGBoost. The input features of the XGBoost model are frequencies of tokens from the vocabulary, excluding any special tokens, along with the age of patients in their first and last visits (5).

LSTM. This is a standard BiLSTM model (19).

CEHR-BERT. An adaptation of the BERT architecture for EHR that introduced the idea of incorporating temporal information using time embeddings and special time interval tokens (10; 38). CEHR-BERT outperformed prior clinical BERT adaptations in various tasks including patient mortality, hospital readmission, and several disease diagnoses (38).

BigBird Transformer. The BigBird Transformer, a variant of the BERT model, employs a form of attention known as block sparse attention (10; 61; 51; 14). This modification allows for more efficient memory usage, facilitating the processing of longer EHR sequences. Here, we use a vanilla BigBird model with a context length of 2048 tokens, 4x greater than that of the CEHR-BERT model (51; 38).

MultiBird Transformer. The MultiBird Transformer adopts the structural framework of the BigBird model but is trained using MPF, see Section 3.3, which enables a single finetuned model to excel across multiple downstream tasks. This training strategy is compatible with existing BigBird model implementations on Hugging Face, simplifying deployment of trained models (55; 61).

4.3. Main Results

Table 1 presents the performance comparison of baseline models and **EHRMAMBA** on the MIMIC-IV dataset (20; 21). EHRMamba and MultiBird show a significant performance advantage over other models, likely due to their finetuning with MPF, which enhances their ability to integrate insights across multiple tasks. This is especially beneficial for complex tasks such as readmission prediction and tasks with fewer data points, such as Condition 2. There is no substantial performance difference between BigBird and CEHR-BERT, underscoring the efficacy of block

Table 1: Performance comparison of **EHRMAMBA** against baseline models on 6 clinical prediction tasks using the MIMIC-IV dataset. These results demonstrate that **EHRMAMBA** achieves state-of-the-art performance while also offering memory and computational efficiency. An asterisk (*) indicates a significant improvement over the best baseline, with a p-value less than 0.05.

Metric	XGBoost	LSTM	CEHR-BERT	BigBird	MultiBird	**EHRMAMBA**
Mortality						
AUROC	0.956 (0.00)	0.942 (0.003)	0.967 (0.001)	0.965 (0.002)	0.968 (0.001)	**0.976 (0.005)***
AUPRC	0.795 (0.00)	0.771 (0.002)	0.857 (0.001)	0.852 (0.004)	0.863 (0.002)	**0.873 (0.002)***
F1-Score	0.633 (0.00)	0.603 (0.002)	0.751 (0.005)	0.754 (0.004)	0.770 (0.002)	**0.784 (0.002)***
Length of Stay						
AUROC	0.835 (0.00)	0.854 (0.003)	0.915 (0.003)	0.914 (0.002)	0.914 (0.002)	**0.919 (0.002)***
AUPRC	0.728 (0.00)	0.774 (0.004)	0.835 (0.005)	0.837 (0.004)	0.839 (0.002)	**0.849 (0.002)***
F1-Score	0.653 (0.00)	0.685 (0.003)	0.727 (0.002)	0.728 (0.003)	0.731 (0.001)	**0.738 (0.002)***
Readmission						
AUROC	0.677 (0.00)	0.654 (0.006)	0.672 (0.003)	0.672 (0.002)	0.681 (0.003)	**0.682 (0.002)**
AUPRC	0.545 (0.00)	0.536 (0.002)	0.543 (0.003)	0.545 (0.006)	0.565 (0.004)	**0.567 (0.004)**
F1-Score	0.533 (0.00)	0.521 (0.004)	0.531 (0.001)	0.535 (0.002)	0.544 (0.003)	**0.545 (0.003)**
Condition 0: Hypertension						
AUROC	0.872 (0.00)	0.854 (0.004)	0.867 (0.002)	0.874 (0.002)	0.878 (0.001)	**0.888 (0.004)***
AUPRC	0.835 (0.00)	0.825 (0.003)	0.827 (0.003)	0.840 (0.003)	0.844 (0.003)	**0.850 (0.002)***
F1-Score	0.754 (0.00)	0.744 (0.002)	0.752 (0.003)	0.761 (0.003)	0.767 (0.003)	**0.772 (0.001)***
Condition 1: Fluid Disorders						
AUROC	0.884 (0.00)	0.874 (0.003)	0.879 (0.001)	0.881 (0.002)	0.886 (0.002)	**0.891 (0.001)***
AUPRC	0.801 (0.00)	0.787 (0.002)	0.793 (0.001)	0.798 (0.002)	0.805 (0.003)	**0.815 (0.001)***
F1-Score	0.717 (0.00)	0.701 (0.002)	0.719 (0.003)	0.723 (0.003)	0.728 (0.001)	**0.733 (0.002)***
Condition 2: Lipoid Metabolism Disorders						
AUROC	0.91 (0.00)	0.893 (0.002)	0.903 (0.000)	0.906 (0.001)	0.919 (0.002)	**0.928 (0.002***
AUPRC	0.785 (0.00)	0.739 (0.002)	0.745 (0.003)	0.767 (0.001)	0.788 (0.003)	**0.804 (0.003)***
F1-Score	0.724 (0.00)	0.683 (0.001)	0.690 (0.002)	0.703 (0.003)	0.731 (0.002)	**0.746 (0.002)***

sparse attention over global attention. However, Big-Bird does outperform CEHR-BERT in condition prediction tasks, due to its longer context length. Notably, XGBoost struggles with tasks that require capturing temporal and sequential information, as it primarily processes token frequencies. Overall, EHRMamba outperforms MultiBird while also being far more memory and computationally efficient, making it a superior choice for a wide range of EHR modeling objectives. For a detailed comparison of the memory and computational efficiency see Appx.C.3.

4.4. Visualized Case Study

We present a case study of deceased Patient X, with the EHR sequence illustrated in Figure 3.

Interpretability. We use integrated gradients to assess the impact of each token in the EHR sequence on the clinical predictions. This method integrates the gradients of the model's outputs with respect to the embeddings of each input token, from a baseline (such as all zeros) to the actual input (49). Figure 3 shows the average attribution scores for **EHRMAMBA**'s predictions on the mortality task. Our analysis reveals that lab tests have the highest attribution values among all token types, indicating they are the most influential features for the model's mortality predictions. This implies that the model heavily relies on laboratory test results to assess patient risk.

Forecasting. Using **EHRMAMBA**, we forecast the subsequent event tokens in the patient X's EHR sequence given the preceding tokens before a cutoff. Figure 3 contrasts these predictions with actual events. Though predictions may not match exactly, they frequently capture relevant medical concepts. Appendix D offers additional examples and a quantitative evaluation of **EHRMAMBA**'s forecasting performance.
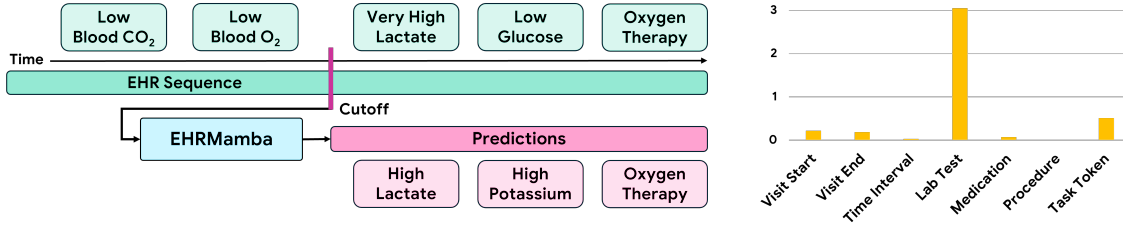
Figure 3: Left) **EhrMamba** predicted tokens. Right) Average token attributions for mortality prediction. Lab tests are the most influential features in **EhrMamba**'s assessment of patient mortality risk.

5. Discussion

Implications and Contributions. **EhrMamba** is a significant advancement in EHR modeling. By addressing the key limitations of existing Transformer models, such as computational inefficiency and the need for separate task-specific finetuning, **EhrMamba** offers a scalable and deployable solution, well-suited for real-world clinical environments. The use of Mamba architecture enables processing longer patient sequences with reduced computational costs, a feasible solution for hospitals with limited resources for advanced predictive modeling. Additionally, the introduction of MPF allows **EhrMamba** to learn multiple clinical tasks concurrently, improving representation learning, generalizability and eliminating the overhead of maintaining multiple models.

Limitations. This study is limited to the MIMIC-IV dataset with EHR data from a single hospital (20; 21). Only three FHIR resources—procedures, medications, and lab tests—are used (32). Moreover, this dataset under represents **EhrMamba**'s ability to handle long EHR sequences. Our model currently does not support multimodal inputs.

Future Directions. We aim to train and assess **EhrMamba** on a larger dataset of patient EHR records from several hospitals in Canada. We plan to include multimodal data inputs such as clinical notes and images, while expanding the context length.

6. Related Works

Predictive modeling using EHR data has been an active area of research, with the goal of driving personalized medicine and improving healthcare quality. Early work by Rajkomar et al. (40) demonstrated the scalability and accuracy of deep learning methods on raw EHR data represented in the FHIR format, outperforming traditional predictive models for tasks like mortality prediction and diagnosis code assignment.

More recent efforts have focused on developing specialized architectures to effectively model the unique characteristics of EHR data. BEHRT (27) was a Transformer model tailored for EHR sequences by incorporating temporal information from diagnoses, medications, and measurements. Similarly, CEHR-BERT (38) extended the BERT architecture to incorporate structured temporal EHR data in the OMOP format, improving performance on various prediction tasks. Med-BERT (41) further adapted the BERT framework for structured EHRs, demonstrating significant improvements in disease prediction.

DuETT (26) introduced a dual event-time Transformer that models both event sequences and their corresponding timings. This temporal modeling approach showed benefits over traditional Transformers on mortality prediction and phenotyping. TransformEHR (60) proposed a denoising sequence-to-sequence encoder-decoder Transformer, pre-trained to predict all future disease diagnoses and outcomes from patient sequences. Foresight (25) used a Generative Pre-trained Transformer for forecasting patient timelines with a variety of medical events, including diagnoses, substance use, and procedures.

7. Conclusion

We introduced **EhrMamba**, a novel EHR foundation model that leverages the Mamba architecture and MPF to overcome the limitations of current Transformer models. **EhrMamba** excels in handling long temporal sequences and learning multiple tasks simultaneously, achieving state-of-the-art performance on six clinical prediction tasks in the MIMIC-IV dataset. Additionally, we open-sourced the **Odyssey** toolkit which supports the development and deployment of EHR models. **EhrMamba** significantly advances EHR modeling, offering a robust, scalable, and generalizable solution for improving patient outcomes and clinical decision-making.

References

[1] Sayed K Ali, Haroon Khan, Jasmit Shah, and K Nadeem Ahmed. An electronic health record system implementation in a resource limited country—lessons learned. *Digital Health*, 9: 20552076231203660, September 2023. ISSN 2055-2076. doi: 10.1177/20552076231203660. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10515600/.

[2] Emmi Antikainen, Joonas Linnosmaa, Adil Umer, Niku Oksala, Markku Eskola, Mark van Gils, Jussi Hernesniemi, and Moncef Gabbouj. Transformers for cardiac patient mortality risk prediction from heterogeneous electronic health records. *Scientific Reports*, 13(1):3517, March 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-30657-1. URL https://www.nature.com/articles/s41598-023-30657-1.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[4] Andrew L. Beam, Benjamin Kompa, Allen Schmaltz, Inbar Fried, Griffin Weber, Nathan Palmer, Xu Shi, Tianxi Cai, and Isaac S. Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 25: 295–306, 2020. ISSN 2335-6936.

[5] 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*, KDD '16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

[6] Youngduck Choi, Chill Yi-I. Chiu, and David Sontag. Learning low-dimensional representations of medical concepts. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2016:41–50, 2016. ISSN 2153-4063.

[7] Shahid Ali Choudhry, Jing Li, Darcy Davis, Cole Erdmann, Rishi Sikka, and Bharat Sutariya. A public-private partnership develops and externally validates a 30-day hospital readmission risk prediction model. *Online Journal of Public Health Informatics*, 5(2), June 2013. ISSN 19472579. doi: 10.5210/ojphi.v5i2.4726. URL http://journals.uic.edu/ojs/index.php/ojphi/article/view/4726.

[8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024.

[9] Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, 6(2):94–98, June 2019. ISSN 2514-6645. doi: 10.7861/futurehosp.6-2-94. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6616181/.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[12] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017.

[13] Gabriel J. Escobar, John D. Greene, Peter Scheirer, Marla N. Gardner, David Draper, and Patricia Kipnis. Risk-adjusting hospital inpatient mortality using automated inpatient, outpatient, and laboratory databases. *Medical Care*, 46(2):232–239, March 2008. ISSN 0025-7079. doi: 10.1097/MLR.0b013e3181589bb6. URL https://journals.lww.com/00005650-200803000-00002.

[14] Adibvafa Fallahpour, Vincent Gureghian, Guillaume J. Filion, Ariel B. Lindner, and Amir Pandi. Codontransformer: a multispecies codon optimizer using context-aware neural networks. *bioRxiv*, 2024. doi: 10.1101/2024.09.13.612903. URL https://www.biorxiv.org/content/early/2024/09/13/2024.09.13.612903.

[15] Emily Getzen, Yucheng Ruan, Lyle Ungar, and Qi Long. Mining for health: A comparison of word embedding methods for analysis of ehrs data. *medRxiv*, 2022. doi: 10.1101/2022.03.05.22271961. URL https://www.medrxiv.org/content/early/2022/03/08/2022.03.05.22271961.

[16] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[17] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[18] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[20] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv, 2023. URL https://physionet.org/content/mimiciv/2.2/.

[21] 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):1, January 2023. ISSN 2052-4463. doi: 10.1038/s41597-022-01899-x. URL https://www.nature.com/articles/s41597-022-01899-x.

[22] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2019.

[23] Hargobind S. Khurana, Robert H. Groves, Michael P. Simons, Mary Martin, Brenda Stoffer, Sherri Kou, Richard Gerkin, Eric Reiman, and Sairam Parthasarathy. Real-time automated sampling of electronic medical records predicts hospital mortality. *The American Journal of Medicine*, 129(7):688–698.e2, July 2016. ISSN 00029343. doi: 10.1016/j.amjmed.2016.02.037. URL https://linkinghub.elsevier.com/retrieve/pii/S0002934316302947.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[25] Zeljko Kraljevic, Dan Bean, Anthony Shek, Rebecca Bendayan, Harry Hemingway, Joshua Au Yeung, Alexander Deng, Alfred Baston, Jack Ross, Esther Idowu, James T Teo, and Richard J B Dobson. Foresight—a generative pretrained transformer for modelling of patient timelines using electronic health records: a retrospective modelling study. *The Lancet Digital Health*, 6(4):e281–e290, April 2024. ISSN 25897500. doi: 10.1016/S2589-7500(24)00025-6. URL https://linkinghub.elsevier.com/retrieve/pii/S2589750024000256.

[26] Alex Labach, Aslesha Pokhrel, Xiao Shi Huang, Saba Zuberi, Seung Eun Yi, Maksims Volkovs, Tomi Poutanen, and Rahul G. Krishnan. Duett: Dual event time transformer for electronic health records, 2023.

[27] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: Transformer for electronic health records. *Scientific Reports*, 10 (1):7155, April 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-62922-y. URL https://doi.org/10.1038/s41598-020-62922-y.

[28] Yikuan Li, Mohammad Mamouei, Gholamreza Salimi-Khorshidi, Shishir Rao, Abdelaali Hassaine, Dexter Canoy, Thomas Lukasiewicz, and Kazem Rahimi. Hi-behrt: Hierarchical transformer-based model for accurate prediction of clinical events using multi-modal longitudinal electronic health records. *IEEE Journal of Biomedical and Health Informatics*, 27(2): 1106–1117, February 2023. ISSN 2168-2194, 2168-2208. doi: 10.1109/JBHI.2022.3224727. URL https://ieeexplore.ieee.org/document/9964038/.

[29] Vincent Liu, Patricia Kipnis, Michael K. Gould, and Gabriel J. Escobar. Length of stay predictions: Improvements through the use of automated laboratory and comorbidity variables. *Medical Care*, 48 (8):739–744, August 2010. ISSN 0025-7079. doi: 10.1097/MLR.0b013e3181e359f3. URL https://journals.lww.com/00005650-201008000-00011.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[31] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation, 2024.

[32] Joshua C Mandel, David A Kreda, Kenneth D Mandl, Isaac S Kohane, and Rachel B Ramoni. Smart on fhir: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908, September 2016. ISSN 1527-974X, 1067-5027. doi: 10.1093/jamia/ocv189. URL https://academic.oup.com/jamia/article/23/5/899/2379865.

[33] Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive ehr time-series pre-training benchmark. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, page 257–278, New York, NY, USA, April 2021. Association for Computing Machinery. ISBN 9781450383592. doi: 10.1145/3450439.

3451877. URL `https://dl.acm.org/doi/10.1145/3450439.3451877`.

[34] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[35] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning, 2023.

[36] Kenney Ng, Steven R. Steinhubl, Christopher deFilippi, Sanjoy Dey, and Walter F. Stewart. Early detection of heart failure using electronic health records: Practical implications for time before diagnosis, data diversity, data quantity, and data density. *Circulation: Cardiovascular Quality and Outcomes*, 9(6): 649–658, November 2016. ISSN 1941-7713, 1941-7705. doi: 10.1161/CIRCOUTCOMES.116.002797. URL `https://www.ahajournals.org/doi/10.1161/CIRCOUTCOMES.116.002797`.

[37] Oanh Kieu Nguyen, Anil N. Makam, Christopher Clark, Song Zhang, Bin Xie, Ferdinand Velasco, Ruben Amarasingham, and Ethan A. Halm. Predicting all-cause readmissions using electronic health record data from the entire hospitalization: Model development and comparison. *Journal of Hospital Medicine*, 11(7):473–480, July 2016. ISSN 1553-5592, 1553-5606. doi: 10.1002/jhm.2568. URL `https://shmpublications.onlinelibrary.wiley.com/doi/10.1002/jhm.2568`.

[38] Chao Pang, Xinzhuo Jiang, Krishna S. Kalluri, Matthew E. Spotnitz, Ruijun Chen, Adler J. Perotte, and Karthik Natarajan. CEHR-BERT: incorporating temporal information from structured EHR data to improve prediction tasks. *CoRR*, abs/2111.08585, 2021. URL `https://arxiv.org/abs/2111.08585`.

[39] Michael Pine, Harmon S. Jordan, Anne Elixhauser, Donald E. Fry, David C. Hoaglin, Barbara Jones, Roger Meimban, David Warner, and Junius Gonzales. Modifying icd-9-cm coding of secondary diagnoses to improve risk-adjustment of inpatient mortality rates. *Medical Decision Making*, 29(1):69–81, January 2009. ISSN 0272-989X, 1552-681X. doi: 10.1177/0272989X08323297. URL `http://journals.sagepub.com/doi/10.1177/0272989X08323297`.

[40] Alvin Rishi Rajkomar, Eyal Oren, Kai Chen, Andrew Dai, Nissan Hajaj, Mila Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, Patrik Per Sundberg, Hector Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin Duggan, Jamie Irvine, Quoc Le, Kurt Litsch, Alex Mossin, Justin Jesada Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, Samuel Volchenboum, Kat Chou, Michael Pearson, Srinivasan Madabushi, Nigam Shah, Atul Butte, Michael Howell, Claire Cui, Greg Corrado, and Jeff Dean. Scalable and accurate deep learning for electronic health records. *npj Digital Medicine*, 2018. URL `https://www.nature.com/articles/s41746-018-0029-1`.

[41] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine*, 4(1):1–13, May 2021. ISSN 2398-6352. doi: 10.1038/s41746-021-00455-y. URL `https://www.nature.com/articles/s41746-021-00455-y`.

[42] Narges Razavian, Jake Marcus, and David Sontag. Multi-task prediction of disease onsets from longitudinal lab tests, 2016.

[43] Michael J. Rothman, Steven I. Rothman, and Joseph Beals. Development and validation of a continuous measure of patient condition using the electronic medical record. *Journal of Biomedical Informatics*, 46 (5):837–848, October 2013. ISSN 15320464. doi: 10.1016/j.jbi.2013.06.011. URL `https://linkinghub.elsevier.com/retrieve/pii/S1532046413000865`.

[44] Maurice Rupp, Oriane Peter, and Thirupathi Pattipaka. *ExBEHRT: Extended Transformer for Electronic Health Records*, page 73–84. Springer Nature Switzerland, 2023. ISBN 9783031395390. doi: 10.1007/978-3-031-39539-0_7. URL `http://dx.doi.org/10.1007/978-3-031-39539-0_7`.

[45] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling, 2024.

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

[47] Ethan Steinberg, Ken Jung, Jason A. Fries, Conor K. Corbin, Stephen R. Pfohl, and Nigam H. Shah. Language models are an effective patient representation learning technique for electronic health record data, 2020.

[48] Hong Sun, Kristof Depraetere, Jos De Roo, Giovanni Mels, Boris De Vloed, Marc Twagirumukiza, and

Dirk Colaert. Semantic processing of ehr data for clinical research. *Journal of Biomedical Informatics*, 58:247–259, December 2015. ISSN 15320464. doi: 10.1016/j.jbi.2015.10.009. URL `https://linkinghub.elsevier.com/retrieve/pii/S1532046415002312`.

[49] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[50] Ying P Tabak, Xiaowu Sun, Carlos M Nunez, and Richard S Johannes. Using electronic health record data to develop inpatient mortality predictive model: Acute laboratory risk of mortality score (alarms). *Journal of the American Medical Informatics Association*, 21(3): 455–463, May 2014. ISSN 1067-5027, 1527-974X. doi: 10.1136/amiajnl-2013-001790. URL `https://academic.oup.com/jamia/article-lookup/doi/10.1136/amiajnl-2013-001790`.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[52] Colin Walsh and George Hripcsak. The effects of data sources, cohort selection, and outcome definition on a predictive model of risk of thirty-day hospital readmissions. *Journal of Biomedical Informatics*, 52:418–426, December 2014. ISSN 1532-0464. doi: 10.1016/j.jbi.2014.08.006. URL `https://www.sciencedirect.com/science/article/pii/S1532046414001877`.

[53] Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. Multitask prompt tuning enables parameter-efficient transfer learning, 2023.

[54] Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting?, 2024.

[55] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[56] Michael Wornow, Yizhe Xu, Rahul Thapa, Birju Patel, Ethan Steinberg, Scott Fleming, Michael A. Pfeffer, Jason Fries, and Nigam H. Shah. The shaky foundations of large language models and foundation models for electronic health records. *npj Digital Medicine*, 6(1):135, July 2023. ISSN 2398-6352.

doi: 10.1038/s41746-023-00879-8. URL `https://www.nature.com/articles/s41746-023-00879-8`.

[57] Zhenbang Wu, Huaxiu Yao, David Liebovitz, and Jimeng Sun. An iterative self-learning framework for medical domain generalization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 54833–54854. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/ac0035c349f3fe8af6a93fe44697b5bd-Paper-Conference.pdf`.

[58] Yang Xiang, Jun Xu, Yuqi Si, Zhiheng Li, Laila Rasmy, Yujia Zhou, Firat Tiryaki, Fang Li, Yaoyun Zhang, Yonghui Wu, Xiaoqian Jiang, Wenjin Jim Zheng, Degui Zhi, Cui Tao, and Hua Xu. Time-sensitive clinical concept embeddings learned from large electronic health records. *BMC Medical Informatics and Decision Making*, 19(S2):58, April 2019. ISSN 1472-6947. doi: 10.1186/s12911-019-0766-3. URL `https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0766-3`.

[59] Xiongxiao Xu, Yueqing Liang, Baixiang Huang, Zhiling Lan, and Kai Shu. Integrating mamba and transformer for long-short range time series forecasting, 2024.

[60] 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. *Nature Communications*, 14 (1):7857, November 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-43715-z. URL `https://www.nature.com/articles/s41467-023-43715-z`.

[61] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences, 2021.

[62] Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019.

Appendix

Appendix A. ODYSSEY Toolkit

The **ODYSSEY** toolkit is designed to support the development and deployment of EHR foundation models. It is publicly available at:

https://github.com/VectorInstitute/odyssey

The toolkit includes 4 major modules:

- **data.** This module includes scripts designed for gathering EHR datasets from HL7 FHIR resources (32; 16). It handles the generation and processing of patient sequences for each clinical task, tokenizing the data, and creating the necessary data splits for model training. Additionally, it provides the dataset class used for training the models.

- **models.** This module offers implementations for various models used in this study, including XGBoost, LSTM, CEHR-BERT, BigBird, MultiBird, and EHRMamba. It also includes various embedding classes essential for the models.

- **evals.** This module includes tools for testing models on different clinical prediction tasks and forecasting. It provides evaluation metrics for both training and testing data, ensuring a thorough assessment of model performance.

- **interp.** This module contains methods for interpreting model decisions. It includes interactive visualization of the attention matrix for Transformer-based models, novel interpretability techniques for EHRMamba, and gradient attribution methods. These tools enhance the transparency and understanding of model decisions.

Appendix B. Data Information

B.1. Dataset

MIMIC-IV is a comprehensive EHR dataset from the Beth Israel Deaconess Medical Center, covering patient data from 2008 to 2019, including records from over 431,000 visits and 180,000 patients (20; 21). The dataset is deidentified in accordance with HIPAA Safe Harbor standards and is accessible to credentialed researchers. MIMIC-IV is organized into several key tables that capture a wide range of clinical data. The most relevant components for our study include:

- **Procedures:** This table lists the medical procedures performed during hospital stays, such as surgeries, diagnostic tests, and therapeutic interventions. For instance, it might include details of a heart surgery or a diagnostic colonoscopy, along with when each procedure was performed.

- **Medications:** This table records medications given to patients, including drug names, dosages, methods of administration (e.g., oral, intravenous), and administration times. For example, it might show that a patient received 500 mg of Amoxicillin orally twice a day.

- **Lab Tests:** This table includes results from laboratory tests, such as blood tests, imaging studies, and other diagnostics. For example, it might list blood glucose levels or MRI results. Continuous lab test results are categorized into five bins, each represented by a unique token.

B.2. Clinical Predictive Tasks

- **Mortality Prediction.** The label is determined by comparing the timestamp of the final recorded event in the patient's last visit to the timestamp of death. If the interval is less than 32 days, the outcome is labeled as 1; otherwise, it is labeled as 0 (50; 13; 23).

- **Length of Stay Prediction.** Sequences are truncated to exclude tokens with timestamps beyond the initial 24 hours of the last visit. The sequence is labeled as 1 if the discharge timestamp exceeds one week; otherwise, it is labeled as 0 (29; 40).

- **Readmission Prediction.** The last visit data is excluded entirely. If the time interval between the last visit and its preceding visit is less than one month, as indicated by the special tokens for time intervals, the sequence is labeled as 1; otherwise, it is labeled as 0 (37; 52; 7).

- **Condition Predictions.** For predicting Condition 0 (Hypertension), Condition 1 (Disorders of fluid, electrolyte, acid-base balance), and Condition 2 (Disorders of lipoid metabolism), the sequence remains unchanged. The label is assigned 1 if the condition is present, 0 otherwise (43; 42; 36).

B.3. Data Split

The MIMIC-IV dataset is split into pretraining (57%), finetuning (28%), and test (15%) sets. The pretraining set is only utilized for MLM or NTP training, without incorporating clinical task labels (33). The finetuning set is used for EHR predictive modeling and has a greater positive ratio of task labels than the original dataset distribution. This set has no overlap with the pretraining set to ensure unbiased few shot learning evaluation. Model performance on clinical tasks and forecasting is assessed using the test set. Detailed dataset statistics are presented in Tables 2 and 3.

B.4. Code and Data Access

The MIMIC-IV dataset is publicly available at (20; 21; 32; 16):

$$\text{https:}$$
$$\text{//physionet.org/content/mimic-iv-fhir/1.0/.}$$

All models were developed and evaluated in this study using the **Odyssey** toolkit.

Appendix C. Model Implementation and Training

Here, we describe the implementation and training of models used in our experiments. All models, except XGBoost and LSTM, were trained using the AdamW optimizer (24; 30) with a linear learning rate scheduler. During the initial 10% of training, the learning rate increased from 5×10^{-7} to 5×10^{-5}, and during the remaining 90%, it decreased back to 5×10^{-5}. While XGBoost was trained on a single NVIDIA T4 16GB GPU, others utilized NVIDIA A100 80GB GPUs.

C.1. Mamba Block

The input tensor $H_{in} \in \mathbb{R}^{b \times l_c \times d}$ to each Mamba block is first normalized using root mean square layer normalization (RMS normalization) (3; 62). After normalization, the tensor is expanded through two linear projections. One projection undergoes a convolution followed by a SiLU activation, denoted by σ, and its output is processed by the discretized SSM to filter relevant information. The other projection is directly passed through a SiLU activation and combined with the outputs from the SSM using a multiplicative gate. This combined output is then passed through

another linear projection and summed with the initial input H_{in}, resulting in the final output tensor $H_{out} \in \mathbb{R}^{b \times l_c \times d}$. (17; 59; 54)

C.2. Time Embeddings

Time embeddings are designed to encode temporal information, which cannot be directly represented by standard procedures. Inspired by Time2Vec approach (22), a Fourier transform is used to decompose a sequence of time points into sine functions. These functions are controlled by learnable parameters that adapt to training data. The embeddings for concept, time, and age are concatenated and projected to the original dimension, generating temporal concept embeddings (38).

C.3. Compute and Memory Efficiency

EHRMamba demonstrates superior memory and computational efficiency compared to MultiBird, a transformer-based model. During pretraining, EHRMamba processed each epoch in 14:29 minutes with a maximum batch size of 44, while MultiBird required 24:05 minutes with a maximum batch size of 32. The efficiency gap widened during fine-tuning, where EHRMamba completed each epoch in 1 hour 17 minutes with a maximum batch size of 64, compared to MultiBird's 4 hours 37 minutes with a maximum batch size of 26. These results highlight **EhrMamba**'s ability to handle larger batch sizes and process data more rapidly, showcasing its enhanced memory utilization and computational efficiency.

Appendix D. Forecasting

To evaluate **EhrMamba**'s predictive performance, we conducted an experiment using both train and test datasets. For each patient sequence, we set a cutoff of 10 tokens before the end and tasked the model with predicting these final 10 tokens. We then compared the predictions to the actual tokens using two metrics: accuracy and cosine similarity of token embeddings. This comparison was performed for the first 1, 2, 5, and 10 predicted tokens. The results are summarized in Table 5.

Additionally, Figure 4 illustrates the comparison between **EhrMamba**'s predicted EHR trajectories and the actual trajectories for six sample patients from the test dataset.

Table 2: Dataset Statistics for MIMIC-IV.

	Pretrain	Finetune	Test
# Patients	93,787	47,447	24,924
# Visits	175,413	137,781	55,589
# Tokens	22,559,083	25,936,531	8,607,890

Table 3: Dataset Statistics for Clinical Predictive Tasks.

	Mortality		Length of Stay		Readmission	
	Finetune	Test	Finetune	Test	Finetune	Test
# Patients	47,447	24,924	42,417	21,595	31,775	11,338
# Visits	137,781	55,589	124,889	49186	122,109	42,003
# Tokens	25,936,531	8,607,890	24,426,766	8,080,243	20,851,581	6,289,346
Task Rate	27%	10%	47%	30%	49%	38%

	Hypertension		Fluid Disorders		Lipoid Metabolism Disorders	
	Finetune	Test	Finetune	Test	Finetune	Test
# Patients	47,447	24,924	47,447	24,924	47,447	24,924
# Visits	137,781	55,589	137,781	55,589	137,781	55,589
# Tokens	20,851,581	8,607,890	20,851,581	8,607,890	20,851,581	8,607,890
Task Rate	50%	43%	49%	33%	30%	24%

Table 4: Architecture and Training Details.

		LSTM	CEHR-BERT	BigBird	MultiBird	EHRMamba
Architecture	Embedding Size	768	768	768	768	768
	Time Embedding Size	32	32	32	32	32
	Visit Order Size	3	3	3	3	3
	Type Vocab Size	9	9	9	9	9
	Context Length	512	512	2048	2048	2048
	# Hidden Layers	6	6	6	6	-
	# Mamba Blocks	-	-	-	-	32
	# Heads	-	8	12	12	-
	Intermediate Size	-	3072	3072	3072	-
	State Size	-	-	-	-	16
	Conv Kernel Size	-	-	-	-	4
	Hidden Size	384	-	-	-	-
Pretrain	Batch Size	32	32	32	32	44
	GPUs	-	4	4	4	4
	Epochs	-	15	15	15	15
	Mask Probability	-	0.15	0.15	0.15	-
	Dropout Probability	0.1	0.1	0.1	0.1	0.1
Finetune	Batch Size	32	32	26	26	64
	GPUs	1	4	4	4	4
	Epochs	5	5	3	6	6
	Dropout Probability	0.1	0.1	0.1	0.1	0.1
	Learning Rate	1×10^{-3}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}

Table 5: **EhrMamba** Forecasting Performance

	Test		Train	
# Tokens	Accuracy	Cosine Sim.	Accuracy	Cosine Sim.
1	0.3620	0.3870	0.3884	0.4057
2	0.3318	0.3470	0.3391	0.3640
5	0.2699	0.2929	0.2753	0.2972
10	0.1991	0.2111	0.2055	0.2201

Figure 4: Forecasting results on six patients of the test dataset.

Appendix E. Notations

Special Tokens	
$[CLS]$	Start of sequence token
$[VS], [VE]$	Start of visit and end of visit tokens
$[REG]$	Register token
$[W_0]$ to $[W_3]$	Week tokens (indicating different weeks)
$[M_0]$ to $[M_{12}]$	Month tokens (indicating different months)
$[LT]$	Long-term time interval token
$[PAD]$	Padding token
$[UNK]$	Unknown token
$[MASK]$	Mask token
Event Tokens and Attributes	
$e_{v,j}$	Event token at visit v, index j
$\tau_{v,j}$	Timestamp of the event $e_{v,j}$
T	Type of event (procedure (P), medication (M), or lab result (L))
A	Age of the patient at the time of the event
S	Visit segment, indicating the separation between adjacent visits
V	Visit order, visit number incrementing for each subsequent visit
$\mathcal{P}$	Absolute position of the event within the sequence
Functions and Mappings	
$f(e_{v,j})$	Function mapping the event space $\mathcal{E}$ to a set of attributes Γ
$\mathcal{C}(v,j)$	Context of the token $e_{v,j}$
$\mathcal{M}$	Set of masked indices for Masked Language Modeling (MLM)
$\mathcal{I}$	Indices for Next Token Prediction (NTP) task
Model Parameters and Objectives	
$\theta, \mathcal{D}$	Model parameters and Dataset
$y, \hat{y}(x;\theta)$	Actual and Predicted clinical outcomes
$\mathcal{L}_i$	Binary cross-entropy loss for task c_i
State Space Models (SSMs)	
Δ	Step size for discretization
A, B, C	Learnable matrices in the State Space Model (SSM)
$h(t)$	Hidden state in the SSM
$\overline{A}, \overline{B}$	Discrete parameters derived from continuous parameters
Embedding Layer	
$E_{v,j}$	Embedding of event token $e_{v,j}$
$E_{\text{concept}}(e_{v,j})$	Concept embedding of event token $e_{v,j}$
$E_{\text{type}}(T_{v,j})$	Token type embedding of event token $T_{v,j}$
$E_{\text{age}}(A_{v,j})$	Age embedding of event token $A_{v,j}$
$E_{\text{time}}(\tau_{v,j})$	Time embedding of event token $\tau_{v,j}$
$E_{\text{segment}}(S_{v,j})$	Segment embedding of event token $S_{v,j}$
$E_{\text{visit order}}(V_{v,j})$	Visit order embedding of event token $V_{v,j}$
$E_{\text{position}}(\mathcal{P}_{v,j})$	Positional embedding of event token $\mathcal{P}_{v,j}$