A DATASET

Our primary dataset, "EHR-OMOP", is sourced from an academic medical center. It contains deidentified longitudinal EHR data formatted according to the Observational Medical Outcomes Partnership Common Data Model (OMOP-CDM) (Sciences & Informatics, 2021). All data is stripped of protected health information and deidentified at the institution level to comply with HIPAA and the Safe Harbor standard. The dataset is stored in a HIPAA-compliant compute environment. All patients included in EHR-OMOP sign a form consenting their records to be included in research purposes like this work. This study was conducted under an institution-wide IRB protocol that makes this deidentified dataset available for research purposes.

We use roughly 2.5M patients from EHR-OMOP for pretraining our models, and hold out 0.5M patients for conducting validation experiments.

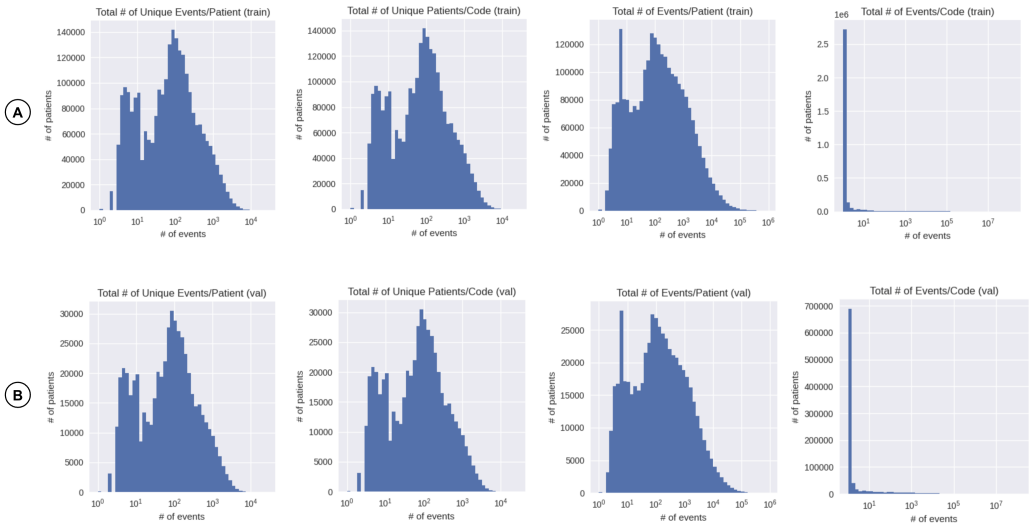


Figure 5: Distributions of patient data from the EHR-OMOP dataset across (A) training and (B) validation splits, showing both event-level and code-level counts. The x-axis is log-scaled to capture the wide range in the number of events per patient, the number of unique patients per code, and the distribution of events associated with each code.

Training Split	**Value**
Overall counts	
Number of events	3,501,210,238
Unique codes	3,144,978
Unique patients	2,567,450
Events per patient	
Minimum	1
Mean	1,364
Median	121
Maximum	890,048
Unique events per patient	
Minimum	1
Mean	237
Median	76
Maximum	26,131

Validation Split	**Value**
Overall counts	
Number of events	749,003,035
Unique codes	881,012
Unique patients	550,305
Events per patient	
Minimum	1
Mean	1,361
Median	121
Maximum	638,708
Unique events per patient	
Minimum	1
Mean	237
Median	76
Maximum	18,561

Table 3: Summary statistics for the EHR-OMOP training (left) and validation (right) splits.

B EVALUATION

B.1 TASKS

For all of our model evaluations, we use 14 binary clinical prediction tasks sourced from the EHRSHOT benchmark (Wornow et al., 2023). The definitions of these tasks are detailed in Appendix Table 4. We also provide label and patient counts in Appendix Table 5 for each task.

Task Name	Task Type	Prediction Time	Time Horizon
Operational Outcomes			
Long Length of Stay	Binary	11:59pm on day of admission	Admission duration
30-day Readmission	Binary	11:59pm on day of discharge	30 days post-discharge
ICU Transfer	Binary	11:59pm on day of admission	Admission duration
Anticipating Lab Test Results			
Thrombocytopenia	Binary	Immediately before result	Next result
Hyperkalemia	Binary	Immediately before result	Next result
Hypoglycemia	Binary	Immediately before result	Next result
Hyponatremia	Binary	Immediately before result	Next result
Anemia	Binary	Immediately before result	Next result
Assignment of New Diagnoses			
Hypertension	Binary	11:59pm on day of discharge	1 year post-discharge
Hyperlipidemia	Binary	11:59pm on day of discharge	1 year post-discharge
Pancreatic Cancer	Binary	11:59pm on day of discharge	1 year post-discharge
Celiac	Binary	11:59pm on day of discharge	1 year post-discharge
Lupus	Binary	11:59pm on day of discharge	1 year post-discharge
Acute MI	Binary	11:59pm on day of discharge	1 year post-discharge

Table 4: The 14 clinical prediction tasks used for evaluating models in this work. *Prediction Time* is the precise time point (up to minute precision) in a patient's timeline when the prediction is made. *Time Horizon* is the length of time considered after the prediction time to determine whether an event occurs, i.e. we only consider a patient "positive" for a new diagnosis of pancreatic cancer if she receives that diagnosis within a year of being discharged. Table reproduced verbatim from (Wornow et al., 2023).

The definitions for each task are provided below (reproduced verbatim from (Wornow et al., 2023)).

Operational Outcomes. These tasks are related to hospital operations. They are defined as follows:

- **Long Length of Stay**: Predict whether a patient's total length of stay during a visit to the hospital will be at least 7 days. The prediction time is at 11:59pm on the day of admission, and visits that last less than one day (i.e. discharge occurs on the same day of admission) are ignored.
- **30-day Readmission**: Predict whether a patient will be re-admitted to the hospital within 30 days after being discharged from a visit. The prediction time is at 11:59pm on the day of admission, and admissions where a readmission occurs on the same day as the corresponding discharge are ignored.
- **ICU Transfer**: Predict whether a patient will be transferred to the ICU during a visit to the hospital. The prediction time is at 11:59pm on the day of admission, and ICU transfers that occur on the same day as admission are ignored.

Anticipating Lab Test Results. These tasks are related to lab value prediction. The prediction time is immediately before the lab result is recorded. They are defined as follows:

- **Thrombocytopenia**: Predict whether a thrombocytopenia lab comes back as normal ($>=150$ 10^9/L) or abnormal (any other reading). We consider all lab results coded as LOINC/LP393218-5, LOINC/LG32892-8, or LOINC/777-3.
- **Hyperkalemia**: Predict whether a hyperkalemia lab comes back as normal ($<=5.5$ mmol/L), or abnormal (any other reading). We consider all lab results coded as LOINC/LG7931-1, LOINC/LP386618-5, LOINC/LG10990-6, LOINC/6298-4, or LOINC/2823-3.

Task Name	Train		Val		Test	
	# Patients (# Positive)	# Labels (# Positive)	# Patients (# Positive)	# Labels (# Positive)	# Patients (# Positive)	# Labels (# Positive)
Operational Outcomes						
Long Length of Stay	1377 (464)	2569 (681)	1240 (395)	2231 (534)	1238 (412)	2195 (552)
30-day Readmission	1337 (164)	2608 (370)	1191 (159)	2206 (281)	1190 (151)	2189 (260)
ICU Transfer	1306 (107)	2402 (113)	1157 (84)	2052 (92)	1154 (75)	2037 (85)
Anticipating Lab Test Results						
Thrombocytopenia	2084 (906)	68776 (22714)	1981 (807)	54504 (17867)	1998 (853)	56338 (19137)
Hyperkalemia	2038 (456)	76349 (1829)	1935 (428)	60168 (1386)	1958 (405)	63653 (1554)
Hypoglycemia	2054 (511)	122108 (1904)	1950 (433)	95488 (1449)	1970 (435)	100568 (1368)
Hyponatremia	2035 (1294)	81336 (23877)	1930 (1174)	64473 (17557)	1956 (1224)	67028 (19274)
Anemia	2092 (1484)	70501 (49028)	1992 (1379)	56224 (38498)	2002 (1408)	58155 (39970)
Assignment of New Diagnoses						
Hypertension	792 (129)	1259 (182)	781 (128)	1247 (175)	755 (129)	1258 (159)
Hyperlipidemia	923 (137)	1684 (205)	863 (140)	1441 (189)	864 (133)	1317 (172)
Pancreatic Cancer	1376 (128)	2576 (155)	1242 (46)	2215 (53)	1246 (40)	2220 (56)
Celiac	1392 (48)	2623 (62)	1252 (8)	2284 (11)	1255 (13)	2222 (21)
Lupus	1377 (79)	2570 (104)	1238 (24)	2225 (33)	1249 (19)	2243 (20)
Acute MI	1365 (130)	2534 (175)	1234 (112)	2176 (145)	1235 (115)	2127 (144)

Table 5: The number of unique patients and total labels for each split of the 14 EHRSHOT tasks evaluated in this work. The prevalence of positive patients/labels is shown in parenthesis. Table reproduced from (Wornow et al., 2023), with updates to reflect the latest version of the EHRSHOT dataset.

- **Hypoglycemia**: Predict whether a hypoglycemia lab comes back as normal (>=3.9 mmol/L) or abnormal (any other reading). We consider all lab results coded as SNOMED/33747003, LOINC/LP416145-3, or LOINC/14749-6.
- **Hyponatremia**: Predict whether a hyponatremia lab comes back as normal (>=135 mmol/L) or abnormal (any other reading). We consider all lab results coded as LOINC/LG11363-5, LOINC/2951-2, or LOINC/2947-0.
- **Anemia**: Predict whether an anemia lab comes back as normal (>=120 g/L) or abnormal (any other reading). We consider all lab results coded as LOINC/LP392452-1.

Assignment of New Diagnoses. These tasks are related to predicting the first diagnosis of a disease. The prediction time is at 11:59pm on the day of discharge from an inpatient visit, and we count any diagnosis that occurs within 365 days post-discharge as a positive outcome. We ignore all discharges in which the patient already has an existing diagnosis of a disease. The tasks are defined as follows:

- **Hypertension**: Predict whether the patient will have her first diagnosis of essential hypertension within the next year. We define hypertension as an occurrence of the code SNOMED/59621000, as well as its children codes in our ontology.
- **Hyperlipidemia**: Predict whether the patient will have her first diagnosis of hyperlipidemia within the next year. We define hyperlipidemia as an occurrence of the code SNOMED/55822004, as well as its children codes in our ontology.
- **Pancreatic Cancer**: Predict whether the patient will have her first diagnosis of pancreatic cancer within the next year. We define pancreatic cancer as an occurrence of the code SNOMED/372003004, as well as its children codes in our ontology.
- **Celiac**: Predict whether the patient will have her first diagnosis of celiac disease within the next year. We define celiac disease as an occurrence of the code SNOMED/396331005, as well as its children codes in our ontology.
- **Lupus**: Predict whether the patient will have her first diagnosis of lupus within the next year. We define lupus as an occurrence of the code SNOMED/55464009, as well as its children codes in our ontology.
- **Acute MI**: Predict whether the patient will have her first diagnosis of an acute myocardial infarction within the next year. We define myocardial infarction as an occurrence of the code SNOMED/57054005, as well as its children codes in our ontology.

B.2 Evaluation Procedure

Each model $m \in \mathcal{M}$ outputs an embedding for each token in its input sequence. Our goal is to aggregate these outputs into a unified representation R_i for each patient i which captures key patterns in their disease trajectory. We will then use this representation R_i to finetune a logistic regression head for our downstream binary classification prediction tasks.

We define two functions. First, we define $S : \mathbf{R}^{n \times d} \to \mathbf{R}^{k \times d}$ to select a subset of k vectors from a set of n vectors. Second, we define $A : \mathbf{R}^{n \times d} \to \mathbf{R}^{d}$ to aggregate a set of n d-dimensional vectors into a single vector. Thus:

$$R_i = A(S(m(\{T_{ik}, ..., T_{i(k+L)}\})))$$

Initial experiments indicated that setting A to simply return the last vector in the sequence (i.e. the most recent token in a patient's timeline) and S to the most recent L tokens in a patient's timeline prior to the timepoint at which the prediction for a task is made performed the best. Thus, we have:

$$R_i = \text{mean}(m(\{T_{i,|T_i|-L}, ..., T_{i|T_i|}\}))$$

Finally, we fit a logistic regression head H on top of these representations in order to apply them to binary prediction tasks. This yields a final prediction P_i of:

$$P_i = H(R_i)$$

which provides the model's estimate for the probability that a specific clinical event occurs within a task-defined window of time for this patient i based on their current representation R_i.

B.3 Patient Statistics

In Appendix Figure 6, we plot the CDF of the number of **raw clinical events** and **tokens** preceding each prediction time for a given task across train/val/test splits. The blue line represents all prediction times, the orange line corresponds to only predictions associated with a positive label. Note that not every clinical event corresponds to a token in our vocabulary, hence many events are dropped during the tokenization process.

B.4 Task-Level Results

We present plots of each model's performance on the 14 individual EHRSHOT tasks in Appendix Figure 9. Additionally, we provide raw numbers on the AUROC differences between each model and the prior SOTA model, CLMBR-t-base, for each task in Appendix Tables 7, 8, 9, 10. We report bootstrapped 95% confidence intervals over 1,000 resamples of the test set for each AUROC difference. Across all context lengths, our results for Mamba are shown in Appendix Table 7, Llama in Appendix Table 8, GPT in Appendix Table 9, and Hyena in Appendix Table 10.

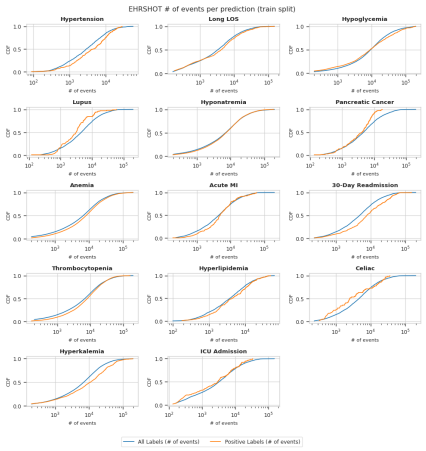
C Model Architectures

In this section, we present the mathematical formulations and detailed architectural descriptions of the four models used in our experiments: GPT, Mamba, Llama, and Hyena.

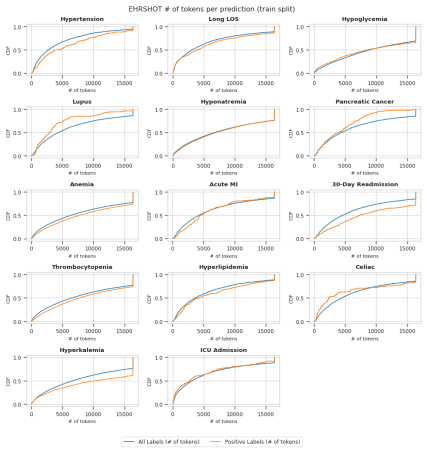
C.1 GPT

GPT (Generative Pre-trained Transformer) is a transformer-based autoregressive model that uses self-attention to process input sequences. (Brown et al., 2020) The main operation is the scaled dot-product attention:

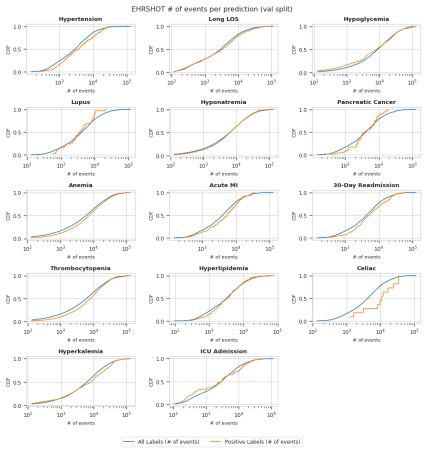
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$



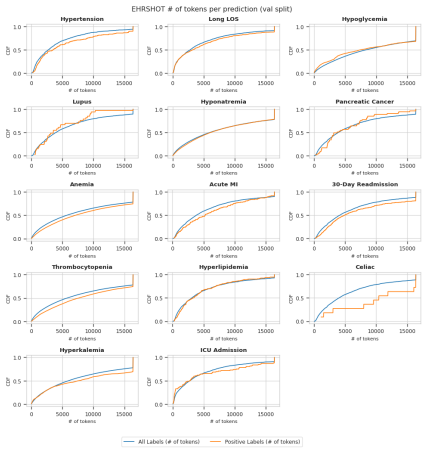
(a) Train (raw clinical events)



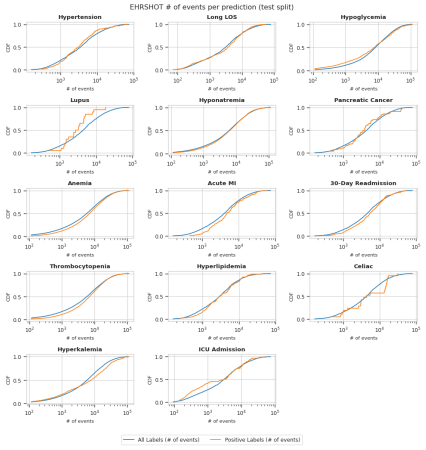
(b) Train (tokens)



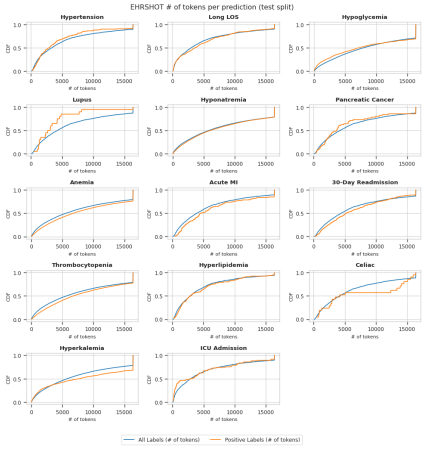
(c) Val (raw clinical events)



(d) Val (tokens)



(e) Test (raw clinical events)



(f) Test (tokens)

Figure 6: For each EHRSHOT task, we plot the CDF of the number of **raw clinical events** (left column) and **tokens** (right column) available to the model when making its prediction. In other words, the number of events/tokens preceding each label's prediction time point. The blue line represents all prediction times, while the orange line represents only predictions associated with a positive label. Note that unlike the raw event counts, all token counts are capped at the maximum context length of the models we test (16k), hence the spike at the end of the CDF.

Here, Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors. The transformer block consists of multi-head attention and a position-wise feed-forward network:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{2}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

where W_i^Q, W_i^K, W_i^V, and W^O are learned projection matrices. After attention, GPT applies a position-wise feed-forward network consisting of two fully connected layers with ReLU activations:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{4}$$

The quadratic complexity of self-attention with respect to input length makes it challenging to scale GPT to long context lengths. In our experiments, we use GPT variants with context lengths up to 4096 tokens.

C.2 LLAMA

Llama is a transformer-based model that shares the core structure of GPT but incorporates optimizations for training efficiency and scalability (Team, 2024). The model uses the same attention mechanism as GPT, but with several architectural modifications, such as an increased hidden state dimension, fewer normalization layers, and relative positional embeddings to improve its performance.

The forward pass for each transformer block in Llama follows the same formulation as GPT, combining self-attention with a feed-forward network:

$$\mathbf{h}_{t+1} = \text{LayerNorm}(\mathbf{h}_t + \text{MultiHead}(\mathbf{h}_t, \mathbf{h}_t, \mathbf{h}_t)) \tag{5}$$

$$\mathbf{h}_{t+2} = \text{LayerNorm}(\mathbf{h}_{t+1} + \text{FFN}(\mathbf{h}_{t+1})) \tag{6}$$

Llama utilizes rotary positional embeddings (RoPE) (Su et al., 2024), which encode relative positional information directly into the self-attention mechanism without requiring absolute positional encodings:

$$\text{RoPE}(q, k, i) = \cos(i\theta)q + \sin(i\theta)k \tag{7}$$

Here, q and k are the query and key vectors, and θ is a frequency parameter. We evaluate Llama on context lengths of up to 4096 tokens.

C.3 MAMBA

Mamba is a state-space model (SSM)-based architecture designed to handle long sequences efficiently. It replaces self-attention with state-space layers, which provide linear scaling with respect to input length. Mamba leverages the continuous-time state-space model to capture long-range dependencies:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t \tag{8}$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t \tag{9}$$

where $\mathbf{x}_t$ is the hidden state, $\mathbf{u}_t$ is the input at time t, $\mathbf{y}_t$ is the output, and A, B, C, and D are learned matrices. This allows Mamba to model long sequences with linear complexity, making it ideal for processing the lengthy and complex event streams in EHR data.

In our experiments, we evaluate Mamba with context lengths of up to 16k tokens. Mamba's efficiency allows it to process long patient histories without the computational overhead of traditional transformer models.

C.4 HYENA

The Hyena architecture introduces an efficient mechanism for handling long sequences by utilizing implicit long convolutions and multiplicative gating (Poli et al., 2023a).

The input sequence is denoted by $\mathbf{x}(t)$, where t represents the sequence position. The convolution operation applied in Hyena can be described by the following equation:

$$\mathbf{y}(t) = \sum_{i=0}^{L-1} \mathbf{h}(i) \cdot \mathbf{x}(t-i)$$

where $\mathbf{x}(t)$ is the input at time step t, $\mathbf{h}(i)$ is the convolution filter of length L, $\mathbf{y}(t)$ is the output at time step t,and L is the length of the filter.

The key difference between Hyena and traditional attention mechanisms is the use of implicit convolutions, which avoid the quadratic complexity of the attention mechanism.

To further enhance the expressivity of the model, Hyena applies multiplicative gating after the convolution operation. This gating mechanism can be expressed as:

$$\mathbf{z}(t) = \sigma(\mathbf{W}_1 \cdot \mathbf{y}(t)) \odot \mathbf{W}_2 \cdot \mathbf{y}(t)$$

where:

- $\mathbf{z}(t)$ is the gated output,
- σ is a non-linear activation function (e.g., sigmoid),
- $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices,
- $\odot$ represents element-wise multiplication.

This combination of implicit long convolutions and multiplicative gating allows the Hyena model to process sequences with log-linear complexity in their length.

D TOKENIZATION

We follow the tokenization strategy used by the CLMBR-t-base model which had achieved the highest average AUROCs on the EHRSHOT benchmark (Wornow et al., 2023). This tokenization strategy is described in detail in (Steinberg et al., 2021).

Given a patient timeline X_i, our goal is to convert it into a sequence of tokens T_i that our models can ingest. Thus, we must map each $X_{ij} = (t_{ij}, c_{ij}, v_{ij})$ to some set of token(s) $T_{ij} = \{T_{ij1}, ..., T_{ijk}\}$ where $T_{ijk} \in \mathbb{T}$.

For encoding the t_{ij} component of each clinical event X_{ij}, we utilize positional encodings based on the token position j, as prior studies have shown minimal benefits from directly embedding absolute time information (Yang et al., 2023).

For handling the v_{ij} component of X_{ij}, we define the following function g to map clinical events to tokens by handling each of the three possible cases for the types of values that v_{ij} can take on separately:

$$g(X_{ij}) = \begin{cases} g_v(c_{ij}) & \text{if } v_{ij} \in \emptyset, \\ g_c(c_{ij}, v_{ij}) & \text{if } v_{ij} \in \mathcal{V}_c, \\ g_n(c_{ij}, v_{ij}) & \text{if } v_{ij} \in \mathcal{V}_n. \end{cases}$$

Thus, the same clinical event (e.g. a lab test for anemia) can be mapped to an arbitrary large set of finer-grained tokens (e.g. one token for all lab tests, one each for mild/moderate/severe, one each for a 10-point scale, etc.).

Following (Steinberg et al., 2021) we choose to employ a deciling strategy for all numerical v_{ij}, and we map each unique categorical v_{ij} to its own token.

Let $D : \mathcal{C} \times \mathcal{V}_n \rightarrow \{x \in \mathbb{Z} \mid 0 \leq x \leq 9\}$ be a function that maps v_{ij} to the decile it belongs to when considering all possible values that c_{ij} is associated with in the training set. And let $G(\cdot)$ be a function that maps its input to some unique integer in the domain of our tokenizer's vocabulary.

Thus, we have that:

$$\begin{aligned} g_v(c_{ij}) &= G(c_{ij}) \\ g_c(c_{ij}, v_{ij}) &= G(c_{ij}, v_{ij}) \\ g_n(c_{ij}, v_{ij}) &= G(c_{ij}, D(c_{ij}, v_{ij})) \end{aligned}$$

Within our dataset, employing this tokenization strategy results in hundreds of thousands of potential unique codes. Many such codes, however, occur very infrequently. Thus, we select the top $k = 39811$ frequently occurring codes, following the same procedure outlined in (Steinberg et al., 2021). In addition, seven special tokens — [BOS], [EOS], [UNK], [SEP], [PAD], [CLS], and [MASK] — are included, resulting in a total vocabulary size of 39818 tokens. This yields an identical vocabulary to the one used by CLMBR-t-base in the original EHRSHOT benchmark (Wornow et al., 2023).

For positional embeddings, we use the default strategies for the various architectures we evaluate – e.g. absolute positional embeddings for GPT, rotary positional embeddings for Llama, none for Hyena beyond the Hyena positional embedding, and none for Mamba.

For completeness, we also evaluate the impact of injecting explicit temporal information into the patient timeline via **Artificial Time Tokens (ATTs)**, as proposed in CEHR-BERT Pang et al., 2021 and used in other works (Pang et al., 2024; Renc et al., 2024). In brief, we create artificial tokens to represent various time intervals (days, weeks, months, etc.) and inject these tokens between consecutive visits to represent the interval of time between them:

$$\text{ATT} = \begin{cases} D_n & \text{if gap } < 7 \text{ days (e.g., } D_1, ..., D_6), \\ W_n & \text{if 7 days} \leq \text{gap} < 28 \text{ days (e.g., } W_1, ..., W_4), \\ M_n & \text{if 28 days} \leq \text{gap} < 365 \text{ days (e.g., } M_1, ..., M_{12}), \\ LT & \text{if gap } \geq 365. \end{cases}$$

Furthermore, to clearly define the start and end of each visit, we enclose each visit V_i with special tokens `VS` (Visit Start) and `VE` (Visit End). This approach allows us to represent a patient timeline as a structured sequence:

$$P = \{\texttt{VS}, v_1, \texttt{VE}, \text{ATT}, \texttt{VS}, v_2, \texttt{VE}, \text{ATT}, \ldots, \texttt{VS}, v_i, \texttt{VE}\}$$

This enhancement directly embeds temporal patterns within the token sequence, providing contextual information about the intervals between clinical events. The results of these models trained using ATT tokens are shown in Appendix Figure 12. The figure shows that this tokenization strategy actually tended to reduce the performance of our models, and our best performing model remains Mamba-16k without ATTs.

E TRAINING

In this section, we describe the training of models used in our experiments. All model base configuration were taken from Huggingface, and can be found uder:

- GPT: https://huggingface.co/openai-community/gpt2
- Hyena: https://huggingface.co/LongSafari/hyenadna-large-1m-seqlen-hf
- Mamba: https://huggingface.co/state-spaces/mamba-130m-hf

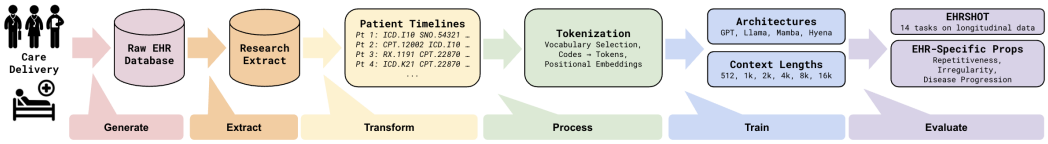


Figure 7: A high-level overview of our experimental pipeline, from data generation to final evaluation results.

- Llama: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Their base configurations were modified to standardize in terms of parameter count to make a fair comparison between them. These configuration changes are shown in Table 6.

Model	Configuration	Value
GPT		
	n positions	{512, 1k, 2k, 4k }
	learning rate	2e-4
	dim model	768
	num layers	12
	num heads	12
	Total Parameters	**116M**
Hyena		
	max seq len	{ 1k, 4k, 8k, 16k }
	learning rate	2e-4
	dim model	768
	num layers	16
	Total Parameters	**125M**
Mamba		
	max seq len	{ 1k, 4k, 8k, 16k }
	learning rate	2e-4
	dim model	768
	num layers	24
	num hidden layers	24
	state size	16
	Total Parameters	**121M**
Llama		
	max position embeddings	{512, 1k, 2k, 4k }
	learning rate	2e-4
	hidden size	768
	intermediate size	2688
	num attention heads	12
	num hidden layers	8
	num key value heads	4
	Total Parameters	**123M**

Table 6: Model configurations used for training. All models are designed to be roughly 120 million parameters. We use the same tokenizer and vocabulary size for all models.

For the pretraining of our models, we randomly sample a patient timeline of length equal to the lesser of the timeline length of the model's context length. To improve training stability and ensure GPU memory optimization, we employed gradient accumulation across multiple batches with a total number of tokens per step of 65,536.

All models were trained using the AdamW optimizer with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\lambda = 0.1$. We performed a hyperparameter sweep over learning rates between $1e-6$ and $1e-3$ for each model architecture before settling on the learning rates shown in Appendix Table 6. We employed a learning rate warm-up for the first 40,000 steps, after which the learning rate decayed to $1e-5$ as training progressed. This approach ensured smooth convergence while avoiding abrupt changes in training dynamics. Perplexity stabilized after one epoch, and we trained all models for 2 billion total tokens.

The training was conducted on a PHI-compliant shared cluster equipped with a heterogeneous mix of GPUs.The majority of experiments in this work were conducted on a set of V100s, with limited access to another 4 NVIDIA H100s and 16 NVIDIA A100s. The use of a secure, PHI-compliant environment ensured that all patient health information remained confidential and protected throughout the training process, adhering to stringent data privacy regulations.

F EHR-SPECIFIC PROPERTY METRICS

We define several metrics for quantifying the specific properties of longitudinal EHR data, such as the irregularity of inter-event time intervals, the repetitiveness of event sequences, and the complexity of tokens due to disease progression. These metrics help us understand the challenges posed by EHR data when used in predictive models.

F.1 REPETITIVENESS

Due to liability, documentation requirements, billing practices, and other administrative processes, EHR data tends to have a high prevalence of "copy-forwarded" information – i.e. data that is copied-and-pasted from one visit to the next (Thornton et al., 2013; Calder et al., 2024; Weis & Levy, 2014). To quantify the level of "copy-forwarding" within a sequence, we calculate the n-gram repetition rate (RR) for each EHR sequence in our dataset using $n = 1, 2, 3, 4$.

We define the n-gram repetition rate as the proportion of n-grams in a given sequence that are repeated at least once. A higher repetition rate means a sequence is more repetitive. Formally, we define the n-gram repetition rate as follows:

$$\text{RR}_n(x) = \frac{\sum_{u \in \mathcal{U}(x)} \mathbb{I}[C(u, x) > 1]}{|\mathcal{U}(x)|}$$

where $\mathcal{U}(\S)$ is the set of unique n-grams in the sequence x and $C(u, x) \in \mathbb{R}$ is the count of occurrences of the n-gram $u \in \mathcal{U}$ in the sequence x. We define $\mathbb{I}[\cdot]$ as the indicator random variable that is 1 if the condition inside the brackets is true, and 0 otherwise.

We calculate n-gram repetition rates for $n = 1, 2, 3, 4$ across all 0.5M patients in our EHR-OMOP validation dataset. In Figure 8, we compare the observed repetition rate in our EHR dataset to the repetition rates observed in the WikiText-103 corpus to demonstrate the higher levels of repetition in EHR sequence data. We repeat our analysis in Appendix Figure 8, but first remove patients with less than 20 total clinical events in order to give a more accurate picture of the level of repetition seen in the timelines of patients with "meaningful" levels of engagement with the healthcare system.

F.2 IRREGULARITY

Irregularity in EHR data arises from uneven time intervals between clinical events for each patient (McDermott et al., 2023). We define three metrics to quantify the irregularity of a given patient's EHR sequence. These metrics help to capture the variability in timing between events, which is critical for models dealing with irregular time intervals.

Standard deviation of inter-event times: Let X_i represent the sequence of clinical events for patient i. Let t_{ij} represent the timestamp of the j-th event in X_i. Then the irregularity $I_\sigma^{(i)}$ of patient i using the standard deviation of inter-event times is given by:

$$\Delta t_{ij} = t_{i(j+1)} - t_{ij}, \quad \forall j \in \{1, \ldots, |X_i| - 1\}$$

$$\mu_i = \frac{1}{|X_i| - 1} \sum_{j=1}^{|X_i|-1} \Delta t_{ij}$$

$$I_\sigma^{(i)} = \sqrt{\frac{1}{|X_i| - 1} \sum_{j=1}^{|X_i|-1} (\Delta t_{ij} - \mu_i)^2}$$

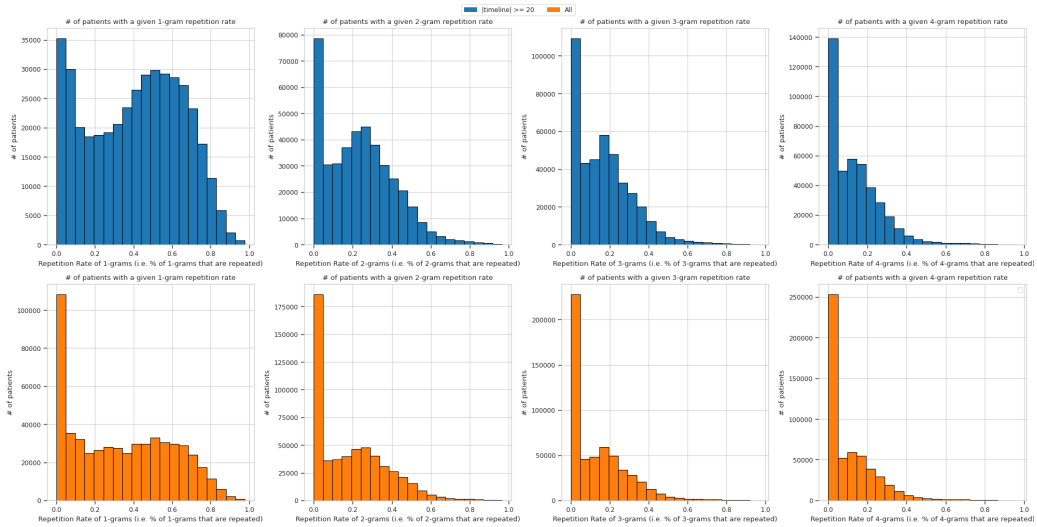


Figure 8: Distribution of n-gram repetition rates across patients in the EHR-OMOP validation set. We repeat our analysis from Figure 3 in the main text (reproduced in the bottom row in orange), but also include a version in which we first filter out all patients with less than 20 total events before generating our plots (top row in blue). This helps to clearly show that patients with "meaningful"-length encounters with the healthcare system tend to have highly repetitive EHR timelines. The x-axis represents the n-gram repetition rate (i.e. percentage of n-grams that are repeated at least once within a patient's EHR), and the y-axis shows the number of patients in each bin.

Mean inter-event time: We can also estimate irregularity as $I_{\mu}^{(i)}$, which represents the mean time between events and is given by:

$$I_{\mu}^{(i)} = \frac{1}{|X_i| - 1} \sum_{j=1}^{|X_i|-1} \Delta t_{ij}$$

Interquartile range (IQR): We can also estimate irregularity as $I_{IQR}^{(i)}$, which represents the interquartile range of the time intervals between events and is given by:

$$I_{IQR}^{(i)} = Q_{75}(\Delta t_{i1}, \ldots, \Delta t_{i(|X_i|-1)}) - Q_{25}(\Delta t_{i1}, \ldots, \Delta t_{i(|X_i|-1)})$$

where $Q_n(\cdot)$ returns the n-th percentile of its arguments.

F.3 Increased Token Complexity Due to Disease Progression

As patients age, their diseases become more complex and varied. Thus, we should expect to see tokens later in a patient's timeline to have higher perlexity than tokens earlier in a patient's timeline. In natural language, the uncertainty of later tokens in a document is reduced by conditioning on all prior tokens, such that later tokens in a prompt typically exhibit substantially lower perplexity than earlier words (Kaplan et al., 2020). We found that this trend did not hold with EHR data, per the experimental set-up described below.

To quantify how the complexity of disease changes over time, we used the median perplexity measured at each token position across patient EHRs. Under our hypothesis of disease progression, later tokens should have higher perplexities, even when conditioning on all prior tokens in a patient's medical history.

Perplexity measures the uncertainty in a model's predictions and is computed as:

$$\text{Perplexity}(x) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i \mid x_{<i})\right)$$

Where x_i is the current token and $P(x_i \mid x_{<i})$ is the predicted probability of the token given the preceding tokens.

More specifically, we start by sampling 20,000 patients from the EHR-OMOP validation set and tokenizing their full timelines. We use this set of patients for all of our subsequent evaluations.

We then select one of our trained models (e.g. Llama with a context length of 512). We use this model to run inference on the full length of each of these 20,000 patients' timelines. This yields a perplexity score for every token. For patient timelines that are longer than the model's context window, we use a sliding window of 32 tokens.

After running inference on all 20,000 patients with this model, we then calculate the median perplexity output by the model at each token positions. We use median rather than mean to reduce the influence of outliers, which we found to be problematic in early testing. We use these median perplexity scores as our official measurement for that token position's perplexity under that model. For our plots, we apply an exponential moving average over the past 250 token positions for smoothing.

F.4 EHRSHOT STRATIFICATION

To stratify model performance on EHRSHOT by the repetitiveness of the underlying patient, we first calculate the 1-gram repetition rate (RR) for each patient in the EHRSHOT test set. After grouping the EHRSHOT test patients by the tasks they belong to, we then stratify the patients within each task by their associated 1-gram RR. We sort patients into 4 quartiles, with Q1 containing patients with the lowest RRs (i.e. the least repetitive patients) and Q4 containg patients with the highest RRs (i.e. the most repetitive patients). For each model and each quartile, we then calculate the average Brier score achieved by that model on all patients within the quartile. This yields one Brier score per quartile per model per task. We chose the Brier score as our performance metric because certain strata exhibited uniform labels, which rendered AUROC calculations infeasible. We repeat this process across all tasks and models.

To obtain a single "Q1" Brier Score for a specific model, we take an unweighted average of the previously calculated mean Brier score for the Q1 patients for each task. We repeat this process for Q2/Q3/Q4 to fill out the full row in the table for a specific model.

For testing the statistical significance of whether two models achieve different Brier scores for the same quartile, we perform 1,000 bootstrap samples over the EHRSHOT test set.

G FEW-SHOT LEARNING ON EHRSHOT

We define k-shot evaluation of a model M on a specific task T as follows:

1. **Training:** For each task T, we sample k positive and k negative examples from the training split of T to train the model M.
2. **Validation:** An additional k positive and k negative examples are sampled from T's validation split to tune hyperparameters for M on T.
3. **Testing:** The best-performing version of M, based on validation results, is evaluated on the entire held-out test split of T. AUROC is recorded as the performance metric.

For tasks where the total number of unique positive examples is fewer than k, all positive examples are included in the training set, and positive examples are randomly resampled until k training examples are achieved.

G.1 EXPERIMENTAL SETUP

We considered values of $k \in \{8, 16, 32, 64, 128\}$ for all 14 EHRSHOT tasks, with one exception: for the *Celiac* prediction task, we limited $k \leq 64$ due to the dataset's constraint of only 62 posi-

tive training examples. This approach ensures fairness in evaluating performance across tasks with varying dataset sizes and class imbalances.

G.2 RESULTS

As shown in Appendix Tables 13, 11, and 12 and Appendix Figure 10, our few-shot learning results indicate that model performance, as measured by AUROC, improves consistently as k increases. Longer-context models, particularly Mamba, demonstrated notable gains even at lower values of k, underscoring their robustness in data-limited scenarios. This trend was consistent across most benchmark tasks, underscoring the utility of long-context architectures in low-resource settings. Our key observations are as follows:

- **Performance Gains with Context Length:** Longer context lengths generally led to better performance, with Mamba models achieving the highest AUROC scores across several k-shot settings, especially at 16,384 tokens.
- **Impact of Few-Shot Sample Size (k):** All models showed improved performance with increasing k, but Mamba and Llama benefited more significantly at higher values of k (64 and 128), consistently outperforming other models across tasks.

H ZERO-SHOT LEARNING ON EHRSHOT

We also evaluate a subset of our models under the **zero-shot** setting, i.e we simply run inference on each model without any finetuning. This offers the practical benefit of not having to train or store any fine-tuned task-specific model heads.

H.1 EXPERIMENTAL SETUP

We follow the procedure outlined in the ETHOS paper (Renc et al., 2024) for making our zero-shot predictions. In brief, we generate 20 synthetic timelines for each patient at the prediction time, measure the percentage of timelines in which the positive event for a task occurs, and then use that percentage as the probability that the patient experiences that positive event. For our zero-shot evaluations, we choose our two strongest models (Mamba and Llama) at their minimum and maximum context lengths, and evaluate them on three representative EHRSHOT tasks – new diagnosis of hypertension, 30-day readmission, and new diagnosis of acute MI.

H.2 RESULTS

As shown in Appendix Table 15, our zero-shot results significantly lag behind the performance of our few-shot and finetuned models. None of the zero-shot models beat the prior SOTA model (CLMBR-t-base) on any of the three tasks evaluated. Additionally, results across context lengths appear mixed. This underscores the importance of finetuning for clinical prediction making, and suggests that our training pipeline is not optimally designed for zero-shot evaluations.

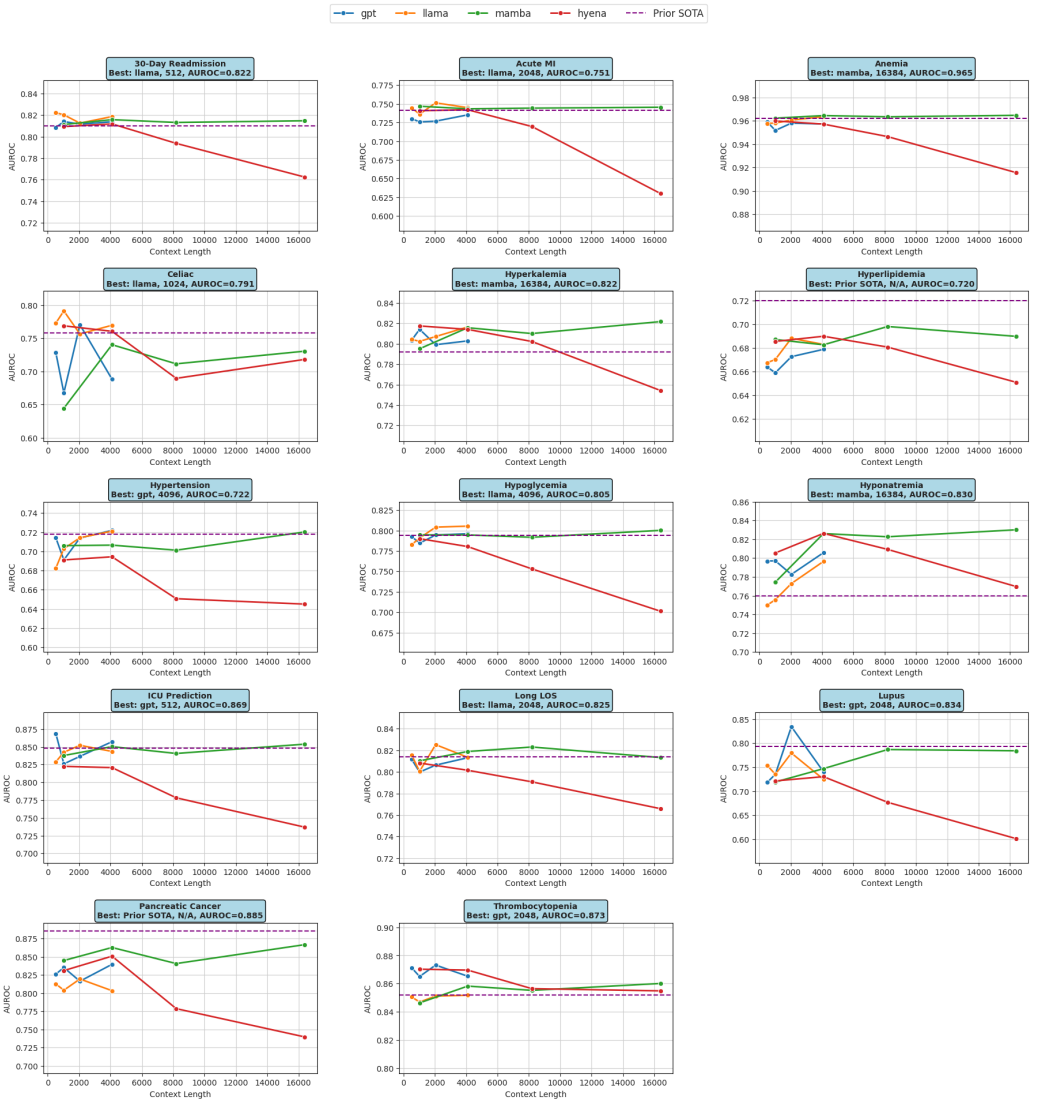


Figure 9: AUROC by context length and architecture across all 14 tasks evaluated from EHRSHOT. The highest scoring model for each task is listed above its plot. Note that the "Prior SOTA" is selected on a task-by-task basis, and thus is not necessarily the same model across plots.

Model	Context Length	Task	Δ over CLMBR-t-base	95% CI	Significant
mamba	1024	ICU Admission	-0.009	(-0.039, 0.019)	
mamba	1024	Long LOS	-0.003	(-0.018, 0.010)	
mamba	1024	30-day Readmission	0.001	(-0.010, 0.013)	
mamba	1024	Anemia	0.000	(-0.001, 0.001)	
mamba	1024	Hyperkalemia	0.003	(-0.006, 0.013)	
mamba	1024	Hypoglycemia	0.001	(-0.011, 0.013)	
mamba	1024	Hyponatremia	0.014	(0.007, 0.022)	✓
mamba	1024	Thrombocytopenia	-0.005	(-0.010, -0.001)	✓
mamba	1024	Acute MI	0.017	(-0.007, 0.040)	
mamba	1024	Celiac	0.102	(-0.076, 0.262)	
mamba	1024	Hyperlipidemia	0.020	(-0.010, 0.050)	
mamba	1024	Hypertension	-0.011	(-0.034, 0.011)	
mamba	1024	Lupus	-0.030	(-0.115, 0.052)	
mamba	1024	Pancreatic Cancer	0.032	(-0.008, 0.071)	
mamba	4096	ICU Admission	0.004	(-0.024, 0.029)	
mamba	4096	Long LOS	0.005	(-0.010, 0.021)	
mamba	4096	30-day Readmission	0.006	(-0.006, 0.017)	
mamba	4096	Anemia	0.002	(0.001, 0.003)	✓
mamba	4096	Hyperkalemia	0.024	(0.014, 0.034)	✓
mamba	4096	Hypoglycemia	0.001	(-0.012, 0.013)	
mamba	4096	Hyponatremia	0.066	(0.057, 0.075)	✓
mamba	4096	Thrombocytopenia	0.007	(0.002, 0.011)	✓
mamba	4096	Acute MI	0.014	(-0.009, 0.036)	
mamba	4096	Celiac	0.198	(0.115, 0.288)	✓
mamba	4096	Hyperlipidemia	0.015	(-0.034, 0.057)	
mamba	4096	Hypertension	-0.010	(-0.033, 0.010)	
mamba	4096	Lupus	-0.003	(-0.091, 0.086)	
mamba	4096	Pancreatic Cancer	0.049	(0.017, 0.081)	✓
mamba	8192	ICU Admission	-0.007	(-0.033, 0.018)	
mamba	8192	Long LOS	0.009	(-0.006, 0.024)	
mamba	8192	30-day Readmission	0.003	(-0.010, 0.016)	
mamba	8192	Anemia	0.001	(0.000, 0.002)	✓
mamba	8192	Hyperkalemia	0.018	(0.008, 0.029)	✓
mamba	8192	Hypoglycemia	-0.002	(-0.014, 0.010)	
mamba	8192	Hyponatremia	0.063	(0.053, 0.072)	✓
mamba	8192	Thrombocytopenia	0.004	(-0.001, 0.008)	
mamba	8192	Acute MI	0.014	(-0.008, 0.036)	
mamba	8192	Celiac	0.173	(0.083, 0.312)	✓
mamba	8192	Hyperlipidemia	0.030	(-0.011, 0.068)	
mamba	8192	Hypertension	-0.016	(-0.036, 0.003)	
mamba	8192	Lupus	0.038	(-0.029, 0.113)	
mamba	8192	Pancreatic Cancer	0.027	(-0.010, 0.062)	
mamba	16384	ICU Admission	0.007	(-0.028, 0.040)	
mamba	16384	Long LOS	0.013	(-0.005, 0.029)	
mamba	16384	30-day Readmission	0.005	(-0.008, 0.017)	
mamba	16384	Anemia	0.002	(0.001, 0.003)	✓
mamba	16384	Hyperkalemia	0.030	(0.019, 0.042)	✓
mamba	16384	Hypoglycemia	0.006	(-0.006, 0.019)	
mamba	16384	Hyponatremia	0.070	(0.061, 0.079)	✓
mamba	16384	Thrombocytopenia	0.008	(0.004, 0.013)	✓
mamba	16384	Acute MI	0.016	(-0.005, 0.036)	
mamba	16384	Celiac	0.194	(0.108, 0.333)	✓
mamba	16384	Hyperlipidemia	0.023	(-0.013, 0.058)	
mamba	16384	Hypertension	0.003	(-0.018, 0.023)	
mamba	16384	Lupus	0.037	(-0.056, 0.132)	
mamba	16384	Pancreatic Cancer	0.053	(0.024, 0.087)	✓

Table 7: Performance of Mamba across all context lengths on the 14 EHRSHOT tasks. The column "Δ over CLMBR-t-base" contains the increase in AUROC relative to CLMBR-t-base, the prior SOTA model on EHRSHOT. The column "95% CI" contains a bootstrapped confidence interval calculated over 1,000 samples of the test set. The column "Significant" contains a checkmark if the CI does not intersect with 0.

Model	Context Length	Task	Δ over CLMBR-t-base	95% CI	Significant
llama	512	ICU Admission	-0.018	(-0.052, 0.015)	
llama	512	Long LOS	0.002	(-0.014, 0.017)	
llama	512	30-day Readmission	0.012	(0.000, 0.024)	✓
llama	512	Anemia	-0.004	(-0.005, -0.003)	✓
llama	512	Hyperkalemia	0.012	(0.004, 0.020)	✓
llama	512	Hypoglycemia	-0.011	(-0.022, 0.001)	
llama	512	Hyponatremia	-0.010	(-0.016, -0.004)	✓
llama	512	Thrombocytopenia	-0.001	(-0.006, 0.004)	
llama	512	Acute MI	0.015	(-0.006, 0.037)	
llama	512	Celiac	0.227	(0.111, 0.356)	✓
llama	512	Hyperlipidemia	0.001	(-0.018, 0.020)	
llama	512	Hypertension	-0.035	(-0.057, -0.012)	✓
llama	512	Lupus	0.005	(-0.084, 0.095)	
llama	512	Pancreatic Cancer	0.001	(-0.044, 0.046)	
llama	1024	ICU Admission	-0.005	(-0.042, 0.032)	
llama	1024	Long LOS	-0.013	(-0.034, 0.005)	
llama	1024	30-day Readmission	0.010	(-0.002, 0.024)	
llama	1024	Anemia	-0.004	(-0.005, -0.003)	✓
llama	1024	Hyperkalemia	0.010	(0.002, 0.019)	✓
llama	1024	Hypoglycemia	-0.003	(-0.014, 0.008)	
llama	1024	Hyponatremia	-0.004	(-0.010, 0.001)	
llama	1024	Thrombocytopenia	-0.005	(-0.009, -0.000)	✓
llama	1024	Acute MI	0.007	(-0.014, 0.029)	
llama	1024	Celiac	0.250	(0.149, 0.359)	✓
llama	1024	Hyperlipidemia	0.003	(-0.016, 0.021)	
llama	1024	Hypertension	-0.014	(-0.033, 0.003)	
llama	1024	Lupus	-0.014	(-0.102, 0.079)	
llama	1024	Pancreatic Cancer	-0.007	(-0.053, 0.037)	
llama	2048	ICU Admission	0.005	(-0.023, 0.033)	
llama	2048	Long LOS	0.014	(-0.003, 0.029)	
llama	2048	30-day Readmission	0.010	(-0.003, 0.023)	
llama	2048	Anemia	-0.002	(-0.003, -0.001)	✓
llama	2048	Hyperkalemia	0.015	(0.005, 0.025)	✓
llama	2048	Hypoglycemia	0.011	(-0.002, 0.023)	
llama	2048	Hyponatremia	0.013	(0.005, 0.020)	✓
llama	2048	Thrombocytopenia	-0.000	(-0.006, 0.004)	
llama	2048	Acute MI	0.022	(-0.001, 0.044)	
llama	2048	Celiac	0.212	(0.083, 0.343)	✓
llama	2048	Hyperlipidemia	0.021	(-0.005, 0.049)	
llama	2048	Hypertension	-0.003	(-0.025, 0.018)	
llama	2048	Lupus	0.031	(-0.049, 0.119)	
llama	2048	Pancreatic Cancer	0.007	(-0.042, 0.053)	
llama	4096	ICU Admission	-0.003	(-0.026, 0.021)	
llama	4096	Long LOS	-0.004	(-0.018, 0.010)	
llama	4096	30-day Readmission	0.013	(0.002, 0.026)	✓
llama	4096	Anemia	0.001	(0.000, 0.002)	✓
llama	4096	Hyperkalemia	0.024	(0.016, 0.033)	✓
llama	4096	Hypoglycemia	0.012	(-0.000, 0.022)	
llama	4096	Hyponatremia	0.036	(0.028, 0.046)	✓
llama	4096	Thrombocytopenia	0.000	(-0.004, 0.005)	
llama	4096	Acute MI	0.015	(-0.008, 0.038)	
llama	4096	Celiac	0.226	(0.097, 0.365)	✓
llama	4096	Hyperlipidemia	0.016	(-0.002, 0.036)	
llama	4096	Hypertension	0.004	(-0.013, 0.021)	
llama	4096	Lupus	-0.023	(-0.097, 0.049)	
llama	4096	Pancreatic Cancer	-0.008	(-0.056, 0.033)	

Table 8: Performance of Llama across all context lengths on the 14 EHRSHOT tasks. The column "Δ over CLMBR-t-base" contains the increase in AUROC relative to CLMBR-t-base, the prior SOTA model on EHRSHOT. The column "95% CI" contains a bootstrapped confidence interval calculated over 1,000 samples of the test set. The column "Significant" contains a checkmark if the CI does not intersect with 0.

Model	Context Length	Task	Δ over CLMBR-t-base	95% CI	Significant
gpt2	512	ICU Admission	0.022	(-0.005, 0.050)	
gpt2	512	Long LOS	-0.002	(-0.017, 0.012)	
gpt2	512	30-day Readmission	-0.002	(-0.013, 0.009)	
gpt2	512	Anemia	-0.003	(-0.004, -0.002)	✓
gpt2	512	Hyperkalemia	0.011	(0.001, 0.021)	✓
gpt2	512	Hypoglycemia	-0.001	(-0.014, 0.012)	
gpt2	512	Hyponatremia	0.037	(0.028, 0.046)	✓
gpt2	512	Thrombocytopenia	0.020	(0.015, 0.025)	✓
gpt2	512	Acute MI	0.001	(-0.022, 0.027)	
gpt2	512	Celiac	0.181	(0.063, 0.295)	✓
gpt2	512	Hyperlipidemia	-0.004	(-0.047, 0.043)	
gpt2	512	Hypertension	-0.003	(-0.021, 0.014)	
gpt2	512	Lupus	-0.031	(-0.110, 0.050)	
gpt2	512	Pancreatic Cancer	0.014	(-0.028, 0.054)	
gpt2	1024	ICU Admission	-0.021	(-0.052, 0.009)	
gpt2	1024	Long LOS	-0.014	(-0.032, 0.004)	
gpt2	1024	30-day Readmission	0.004	(-0.009, 0.015)	
gpt2	1024	Anemia	-0.011	(-0.012, -0.009)	✓
gpt2	1024	Hyperkalemia	0.022	(0.011, 0.033)	✓
gpt2	1024	Hypoglycemia	-0.009	(-0.022, 0.004)	
gpt2	1024	Hyponatremia	0.037	(0.028, 0.046)	✓
gpt2	1024	Thrombocytopenia	0.013	(0.009, 0.019)	✓
gpt2	1024	Acute MI	-0.003	(-0.027, 0.021)	
gpt2	1024	Celiac	0.125	(0.007, 0.274)	✓
gpt2	1024	Hyperlipidemia	-0.008	(-0.053, 0.036)	
gpt2	1024	Hypertension	-0.026	(-0.049, -0.005)	✓
gpt2	1024	Lupus	-0.016	(-0.090, 0.062)	
gpt2	1024	Pancreatic Cancer	0.022	(-0.009, 0.050)	
gpt2	2048	ICU Admission	-0.010	(-0.040, 0.021)	
gpt2	2048	Long LOS	-0.008	(-0.022, 0.006)	
gpt2	2048	30-day Readmission	0.002	(-0.011, 0.014)	
gpt2	2048	Anemia	-0.004	(-0.005, -0.003)	✓
gpt2	2048	Hyperkalemia	0.007	(-0.003, 0.017)	
gpt2	2048	Hypoglycemia	0.001	(-0.013, 0.013)	
gpt2	2048	Hyponatremia	0.023	(0.015, 0.029)	✓
gpt2	2048	Thrombocytopenia	0.021	(0.016, 0.027)	✓
gpt2	2048	Acute MI	-0.003	(-0.030, 0.024)	
gpt2	2048	Celiac	0.227	(0.037, 0.433)	✓
gpt2	2048	Hyperlipidemia	0.005	(-0.014, 0.025)	
gpt2	2048	Hypertension	-0.002	(-0.021, 0.017)	
gpt2	2048	Lupus	0.085	(0.005, 0.165)	✓
gpt2	2048	Pancreatic Cancer	0.004	(-0.032, 0.037)	
gpt2	4096	ICU Admission	0.011	(-0.021, 0.044)	
gpt2	4096	Long LOS	-0.001	(-0.014, 0.014)	
gpt2	4096	30-day Readmission	0.004	(-0.009, 0.015)	
gpt2	4096	Anemia	-0.005	(-0.006, -0.004)	✓
gpt2	4096	Hyperkalemia	0.011	(0.001, 0.021)	✓
gpt2	4096	Hypoglycemia	0.003	(-0.011, 0.015)	
gpt2	4096	Hyponatremia	0.046	(0.036, 0.055)	✓
gpt2	4096	Thrombocytopenia	0.014	(0.009, 0.018)	✓
gpt2	4096	Acute MI	0.006	(-0.022, 0.033)	
gpt2	4096	Celiac	0.149	(0.041, 0.278)	✓
gpt2	4096	Hyperlipidemia	0.012	(-0.018, 0.043)	
gpt2	4096	Hypertension	0.004	(-0.015, 0.024)	
gpt2	4096	Lupus	-0.008	(-0.095, 0.088)	
gpt2	4096	Pancreatic Cancer	0.027	(-0.008, 0.062)	

Table 9: Performance of GPT across all context lengths on the 14 EHRSHOT tasks. The column "Δ over CLMBR-t-base" contains the increase in AUROC relative to CLMBR-t-base, the prior SOTA model on EHRSHOT. The column "95% CI" contains a bootstrapped confidence interval calculated over 1,000 samples of the test set. The column "Significant" contains a checkmark if the CI does not intersect with 0.

Model	Context Length	Task	Δ over CLMBR-t-base	95% CI	Significant
hyena	1024	ICU Admission	-0.026	(-0.064, 0.013)	
hyena	1024	Long LOS	-0.006	(-0.020, 0.011)	
hyena	1024	30-day Readmission	-0.001	(-0.012, 0.010)	
hyena	1024	Anemia	-0.002	(-0.003, -0.001)	✓
hyena	1024	Hyperkalemia	0.026	(0.015, 0.036)	✓
hyena	1024	Hypoglycemia	-0.004	(-0.015, 0.008)	
hyena	1024	Hyponatremia	0.045	(0.036, 0.055)	✓
hyena	1024	Thrombocytopenia	0.019	(0.014, 0.024)	✓
hyena	1024	Acute MI	0.011	(-0.015, 0.038)	
hyena	1024	Celiac	0.224	(0.095, 0.367)	✓
hyena	1024	Hyperlipidemia	0.018	(-0.000, 0.037)	
hyena	1024	Hypertension	-0.026	(-0.053, -0.003)	✓
hyena	1024	Lupus	-0.026	(-0.116, 0.055)	
hyena	1024	Pancreatic Cancer	0.019	(-0.022, 0.060)	
hyena	4096	ICU Admission	-0.026	(-0.058, 0.004)	
hyena	4096	Long LOS	-0.012	(-0.030, 0.006)	
hyena	4096	30-day Readmission	0.002	(-0.012, 0.013)	
hyena	4096	Anemia	-0.005	(-0.006, -0.004)	✓
hyena	4096	Hyperkalemia	0.022	(0.013, 0.033)	✓
hyena	4096	Hypoglycemia	-0.013	(-0.027, 0.001)	
hyena	4096	Hyponatremia	0.066	(0.056, 0.078)	✓
hyena	4096	Thrombocytopenia	0.018	(0.013, 0.023)	✓
hyena	4096	Acute MI	0.013	(-0.013, 0.040)	
hyena	4096	Celiac	0.216	(0.077, 0.370)	✓
hyena	4096	Hyperlipidemia	0.023	(-0.012, 0.057)	
hyena	4096	Hypertension	-0.023	(-0.050, 0.002)	
hyena	4096	Lupus	-0.019	(-0.110, 0.056)	
hyena	4096	Pancreatic Cancer	0.038	(-0.011, 0.092)	
hyena	8192	ICU Admission	-0.069	(-0.106, -0.032)	✓
hyena	8192	Long LOS	-0.023	(-0.041, -0.004)	✓
hyena	8192	30-day Readmission	-0.017	(-0.033, -0.002)	✓
hyena	8192	Anemia	-0.016	(-0.018, -0.014)	✓
hyena	8192	Hyperkalemia	0.010	(0.000, 0.022)	✓
hyena	8192	Hypoglycemia	-0.041	(-0.056, -0.025)	✓
hyena	8192	Hyponatremia	0.049	(0.039, 0.059)	✓
hyena	8192	Thrombocytopenia	0.005	(-0.001, 0.010)	
hyena	8192	Acute MI	-0.009	(-0.038, 0.022)	
hyena	8192	Celiac	0.154	(-0.013, 0.352)	
hyena	8192	Hyperlipidemia	0.014	(-0.026, 0.052)	
hyena	8192	Hypertension	-0.066	(-0.108, -0.030)	✓
hyena	8192	Lupus	-0.073	(-0.189, 0.025)	
hyena	8192	Pancreatic Cancer	-0.033	(-0.088, 0.018)	
hyena	16384	ICU Admission	-0.110	(-0.147, -0.075)	✓
hyena	16384	Long LOS	-0.048	(-0.068, -0.029)	✓
hyena	16384	30-day Readmission	-0.048	(-0.067, -0.026)	✓
hyena	16384	Anemia	-0.047	(-0.051, -0.043)	✓
hyena	16384	Hyperkalemia	-0.038	(-0.054, -0.023)	✓
hyena	16384	Hypoglycemia	-0.093	(-0.109, -0.075)	✓
hyena	16384	Hyponatremia	0.010	(-0.002, 0.021)	
hyena	16384	Thrombocytopenia	0.003	(-0.005, 0.011)	
hyena	16384	Acute MI	-0.100	(-0.145, -0.053)	✓
hyena	16384	Celiac	0.176	(0.029, 0.318)	✓
hyena	16384	Hyperlipidemia	-0.016	(-0.069, 0.034)	
hyena	16384	Hypertension	-0.071	(-0.125, -0.023)	✓
hyena	16384	Lupus	-0.145	(-0.268, -0.017)	✓
hyena	16384	Pancreatic Cancer	-0.073	(-0.148, 0.006)	

Table 10: Performance of Hyena across all context lengths on the 14 EHRSHOT tasks. The column "Δ over CLMBR-t-base" contains the increase in AUROC relative to CLMBR-t-base, the prior SOTA model on EHRSHOT. The column "95% CI" contains a bootstrapped confidence interval calculated over 1,000 samples of the test set. The column "Significant" contains a checkmark if the CI does not intersect with 0.

Model	Context Length	k					
		8	16	32	64	128	All
gpt2	512	**0.661**	**0.714**	**0.747**	**0.779**	**0.794**	**0.830**
gpt2	1024	0.634	0.697	0.732	0.758	0.774	0.813
gpt2	2048	0.654	0.704	0.743	0.771	0.792	0.818
gpt2	4096	0.657	0.706	0.742	0.769	0.791	0.828
llama	512	0.672	**0.716**	0.741	0.767	0.786	0.822
llama	1024	0.662	0.707	0.737	0.769	0.788	0.821
llama	2048	**0.674**	0.714	**0.757**	**0.784**	0.799	**0.833**
llama	4096	0.665	0.709	0.756	0.782	**0.800**	0.826
mamba	1024	0.668	0.719	0.745	0.774	0.786	0.820
mamba	4096	0.681	0.730	0.754	0.784	0.796	0.828
mamba	8192	0.676	0.728	0.753	0.782	0.800	0.826
mamba	16384	**0.685**	**0.734**	**0.761**	**0.791**	**0.804**	**0.831**
hyena	1024	**0.655**	**0.705**	**0.739**	**0.761**	**0.778**	**0.813**
hyena	4096	0.631	0.681	0.725	0.747	0.773	0.811
hyena	8192	0.622	0.669	0.698	0.727	0.750	0.788
hyena	16384	0.587	0.629	0.651	0.676	0.705	0.755

Table 11: **Few-Shot Evaluation:** Average AUROC score for each model and context length across all *Operational Outcomes* tasks and k-shot settings. The highest AUROC across all models for each k is **bolded underlined**, and the maximum value within each model across context lengths for each k is **bolded**.

Model	Context Length	k					
		8	16	32	64	128	All
gpt2	512	0.603	0.634	0.670	0.695	0.713	0.730
gpt2	1024	0.610	0.644	0.672	0.691	0.711	0.719
gpt2	2048	**0.621**	**0.654**	**0.684**	**0.709**	**0.726**	**0.756**
gpt2	4096	0.616	0.642	0.678	0.700	0.722	0.734
llama	512	0.606	0.635	0.665	0.687	0.721	0.739
llama	1024	0.615	0.644	0.670	0.692	0.708	0.740
llama	2048	**0.624**	**0.653**	0.675	0.694	**0.728**	**0.751**
llama	4096	0.621	0.646	**0.679**	**0.695**	0.721	0.741
mamba	1024	0.628	0.652	0.682	0.698	0.716	0.725
mamba	4096	0.630	0.658	0.689	0.704	0.726	0.747
mamba	8192	0.633	0.657	0.690	0.706	0.723	0.747
mamba	16384	**0.647**	**0.668**	**0.698**	**0.711**	**0.732**	**0.756**
hyena	1024	**0.621**	**0.651**	**0.682**	**0.697**	**0.717**	0.740
hyena	4096	0.608	0.638	0.666	0.680	0.709	**0.745**
hyena	8192	0.585	0.608	0.638	0.657	0.671	0.699
hyena	16384	0.540	0.553	0.578	0.597	0.636	0.664

Table 12: **Few-Shot Evaluation:** Average AUROC score for each model and context length across all *Assignment of New Diagnoses* tasks and k-shot settings. The highest AUROC across all models for each k is **bolded underlined**, and the maximum value within each model across context lengths for each k is **bolded**.

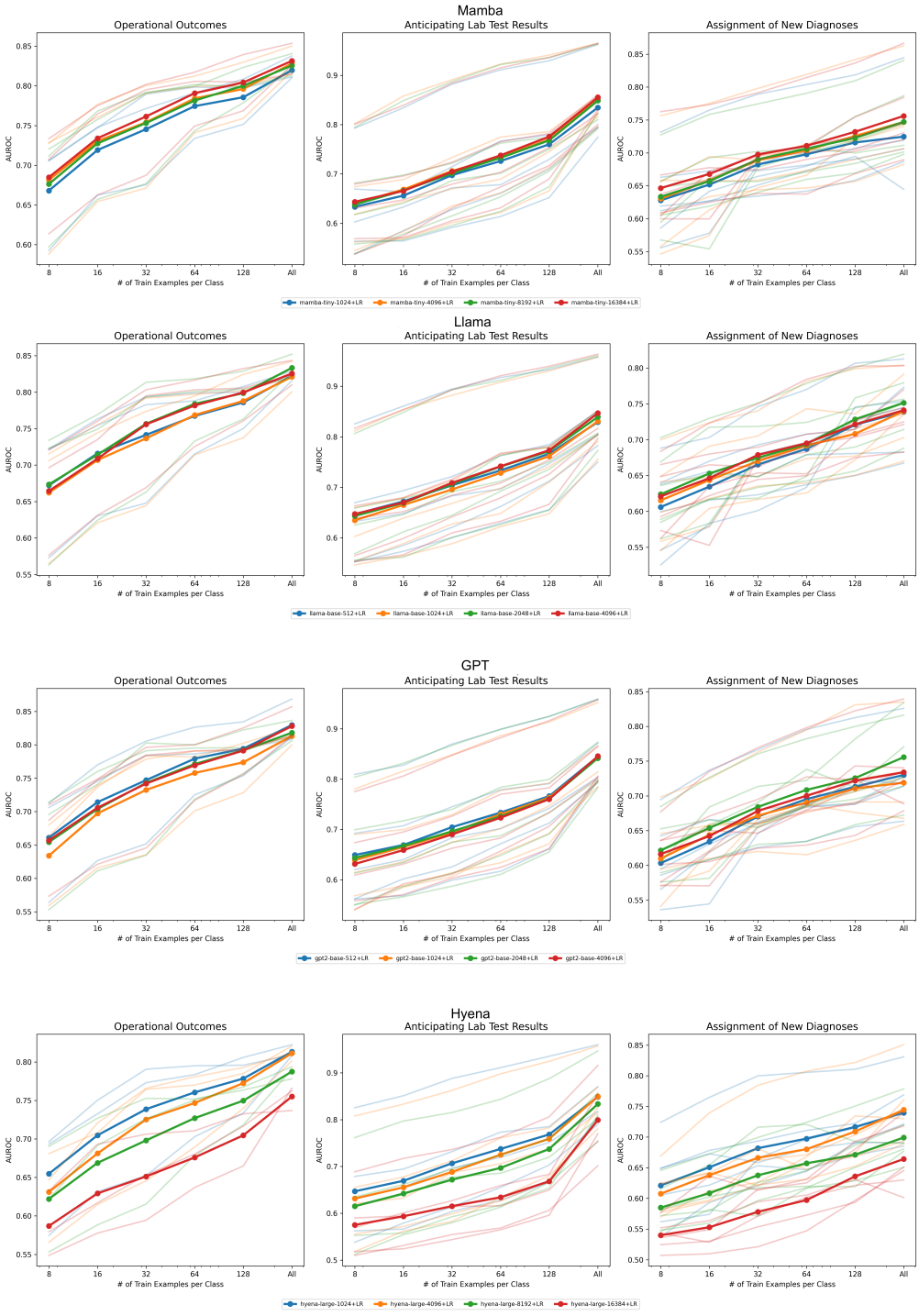


Figure 10: **Few-Shot Evaluation:** Average AUROC scores for each model and context length across all few-shot settings, aggregated for each EHRSHOT clinical prediction task group: *Operational Outcomes*, *Anticipating Lab Test Results*, and *Assignment of New Diagnoses*. Each row is a different model (from top to bottom: Mamba, Llama, GPT, Hyena) and each column is a task group. The x-axis shows the number of few-shot examples (k-shot), while the y-axis displays AUROC. Each line represents a different context length. Solid lines are AUROCs average across all subtasks within a task group, while lighter lines are the few-shot results for each individual subtask.

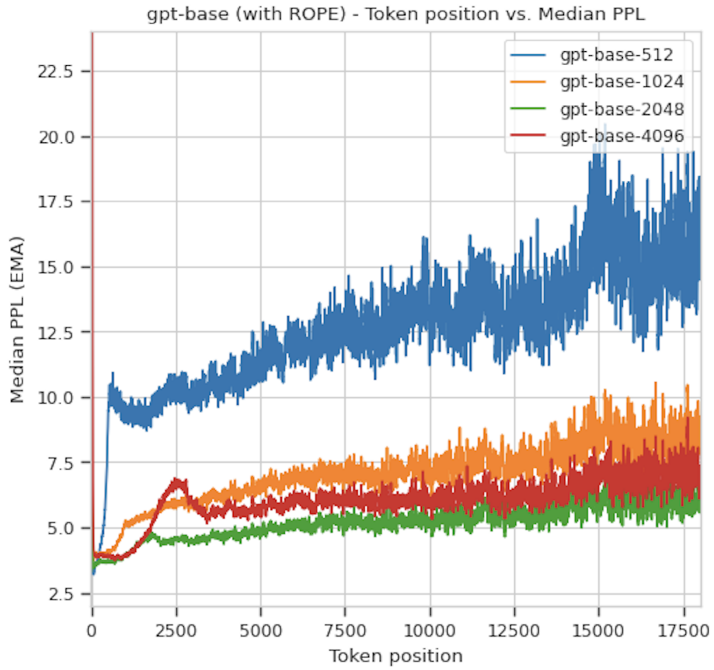


Figure 11: Reproduction of Figure 4 for the GPT architecture, but with rotary positional embeddings (ROPE) instead of absolute positional embeddings. All other aspects of the GPT architecture are kept the same. With ROPE, the perplexity curves appear more stable and do not exhibit the 10+ point perplexity spikes seen in Figure 4, but still mirror the trend of increased perplexity with increased sequence length.

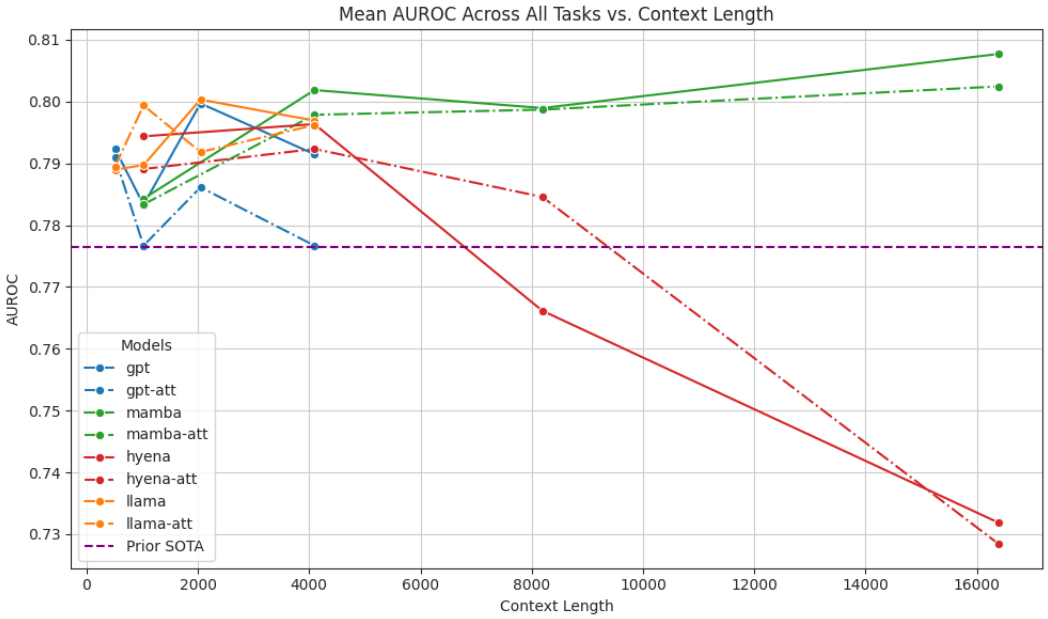


Figure 12: Reproduction of Figure 1b, but with models trained using **Artificial Time Tokens (ATTs)** (as defined in CEHR-BERT (Pang et al., 2021)) shown in dotted lines, and models trained without ATTs in solid lines. Overall, we see better performance without using ATT tokens. While the dotted lines closely follow the solid lines for Mamba and Hyena, the transformer models appear to have less stable performance at smaller contexts, potentially due to the injection of more tokens within each patient's timeline.

Model	Context Length	k					
		8	16	32	64	128	All
gpt2	512	**<u>0.649</u>**	**0.669**	**0.704**	**0.733**	**<u>0.766</u>**	**0.845**
gpt2	1024	0.639	0.665	0.694	0.730	0.763	0.843
gpt2	2048	0.643	0.667	0.696	0.726	0.761	0.841
gpt2	4096	0.631	0.659	0.690	0.723	0.760	**0.845**
llama	512	**0.647**	**<u>0.672</u>**	0.704	0.733	0.767	0.829
llama	1024	0.635	0.665	0.696	0.728	0.762	0.831
llama	2048	0.643	0.669	0.707	0.741	0.772	0.839
llama	4096	**0.647**	0.670	**<u>0.709</u>**	**<u>0.742</u>**	**0.773**	**0.847**
mamba	1024	0.633	0.656	0.698	0.726	0.760	0.835
mamba	4096	0.640	**0.669**	**0.706**	0.734	0.770	0.852
mamba	8192	0.638	0.666	0.701	0.733	0.768	0.849
mamba	16384	**0.644**	0.666	0.705	**0.738**	**<u>0.776</u>**	**<u>0.855</u>**
hyena	1024	**0.647**	**0.669**	**0.707**	**0.737**	**0.768**	0.849
hyena	4096	0.632	0.655	0.688	0.725	0.759	**0.850**
hyena	8192	0.615	0.642	0.672	0.697	0.737	0.833
hyena	16384	0.575	0.594	0.615	0.634	0.668	0.799

Table 13: **Few-Shot Evaluation:** Average AUROC score for each model and context length across all *Anticipating Lab Test Results* tasks and k-shot settings. The highest AUROC across all models for each k is **<u>bolded underlined</u>**, and the maximum value within each model across context lengths for each k is **bolded**.

Metric	Model	Context Length	Q1	Q2	Q3	Q4
Repetitiveness (1-gram RR)	Mamba	1k	0.0644	0.0737	0.0744	0.0790
		16k	**0.0605**	**0.0670**	**0.0700**	**0.0746**
	Llama	512	0.0640	0.0710	0.0743	0.0792
		4k	**0.0627**	**0.0687**	**0.0721**	**0.0770**
	GPT	512	0.0619	0.0691	0.0710	0.0765
		4k	0.0643	0.0692	0.0711	0.0765
	Hyena	1k	0.0636	0.0681	0.0718	0.0776
		16k	0.0733	0.0759	0.0780	0.0822
	CLMBR-t-base	512	0.0647	0.0719	0.0751	0.0805
Irregularity (Standard Deviation)	Mamba	1k	0.0693	0.0729	0.0731	0.0764
		16k	**0.0641**	**0.0678**	**0.0679**	**0.0723**
	Llama	512	0.0694	0.0730	0.0713	0.0749
		4k	**0.0664**	**0.0705**	**0.0694**	0.0740
	GPT	512	0.0654	0.0693	0.0703	0.0736
		4k	0.0653	0.0699	0.0701	0.0759
	Hyena	1k	0.0666	0.0702	0.0692	0.0751
		16k	0.0698	0.0755	0.0788	0.0853
	CLMBR-t-base	512	0.0683	0.0741	0.0721	0.0777

Table 14: Comparison of average Brier scores for all models across all 14 EHRSHOT tasks. Patients are bucketed by repetitiveness (top) and irregularity (bottom). Q1/Q2/Q3/Q4 are the 1st through 4th quartiles of patients ranked by each metric. For example, Q1 contains the least repetitive / least irregular patients while Q4 contains the most repetitive / most irregular patients. **Bolded** values show a statistically significant win rate of at least 50% of the longer context model over the shorter context model at a specific quartile. This is identical to Table 2, but with all models shown.

Model	Context Length	AUROC
Hypertension		
CLMBR-t-base	512	**0.718**
Mamba	1024	0.660
Llama	512	0.642
Llama	4096	0.609
Mamba	16384	0.563
30-day Readmission		
CLMBR-t-base	512	**0.810**
Mamba	1024	0.720
Llama	4096	0.710
Llama	512	0.705
Mamba	16384	0.643
Acute MI		
CLMBR-t-base	512	**0.729**
Mamba	16384	0.531
Mamba	1024	0.525
Llama	4096	0.52
Llama	512	0.51

Table 15: **Zero-Shot Evaluation:** AUROC scores for each model and context length for zero-shot evaluations across three EHRSHOT clinical prediction tasks. The zero-shot evaluations followed the procedure outlined in (Renc et al., 2024). Namely, 20 synthetic timelines were generated for each patient at each prediction timepoint. The probability that a patient experienced a positive event was calculated as the percentage of generated timelines that contained that positive event within the appropriate time horizon as defined by the relevant task.