LARGE LANGUAGE MODELS FOR MULTIMODAL IRREGULAR TIME SERIES WITH INFORMATIVE SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal irregular time series (MITS) consist of asynchronous observations from heterogeneous numerical and textual channels, such as lab measurements and clinical notes in electronic health records (EHR). LLMs are natural candidates for processing such heterogeneous data, yet existing MITS models rely on specialized fusion mechanisms rather than leveraging LLMs for unified MITS processing, and LLM-based approaches for irregular time series address purely numerical observations. Moreover, no prior work has examined whether LLMs can learn to use informative sampling patterns. We introduce **MILM** (**M**ultimodal **I**rregular time series **L**anguage **M**odel), which serializes MITS into a time-ordered sequence of time-channel-value triplets and fine-tunes an LLM through a two-stage strategy for MITS classification. The first stage trains on value-redacted MITS to predict from sampling patterns alone, and the second stage trains on full MITS to jointly model sampling patterns and observed values. MILM achieves the best average rank of 1.1 against 15 baselines on 4 EHR datasets, improving average precision by up to 3.1 points over the strongest baseline. The two-stage strategy also outperforms directly fine-tuning on full MITS in a single stage (MILM-Direct), even when the single-stage baseline receives the same total number of training epochs. Value redaction evaluations confirm that sampling patterns carry predictive signal and that MILM learns to exploit them. Under the value pending evaluation we introduce, where some values are unavailable at prediction time, MILM remains ahead of MILM-Direct when the affected observations are dropped, and its lead widens when their time and channel are shown. Our code is available at `https://anonymous.4open.science/r/milm-7750`.

1 INTRODUCTION

Multimodal irregular time series (MITS) (Chang et al., 2025) consist of asynchronous and irregularly sampled observations from heterogeneous channels spanning numerical and textual modalities. MITS are prevalent in healthcare (Zhang et al., 2023b), finance (Koval et al., 2025), and system monitoring (Zhang et al., 2024a), among others. For example, in electronic health records (EHR) (Johnson et al., 2023c; 2024; Pollard et al., 2018; 2019), a patient's stay comprises lab measurements and clinical notes recorded irregularly over time. Unlike regularly sampled time series, MITS are governed by two processes: a data process that generates the values and a sampling process that determines when and which channels are observed. The timing and channel of each observation therefore carry predictive signal beyond its value.

In this work, we focus on MITS classification motivated by clinical prediction tasks (Zhang et al., 2023b; Khadanga et al., 2019; Han et al., 2024). A canonical example is in-hospital mortality prediction, where a model processes a patient's lab measurements and clinical notes to predict whether the patient will survive their hospital admission. Accurate prediction requires interpreting numerical lab values alongside clinical narratives, while also exploiting when and which lab tests are ordered and clinical notes are authored as informative signals about the patient's evolving condition. LLMs are natural candidates for processing such heterogeneous observations, given their extensive pretrained knowledge spanning both textual and numerical domains. Existing MITS models (Zhang et al., 2023b; Khadanga et al., 2019; Han et al., 2024) focus on specialized multimodal fusion mechanisms, leaving unified LLM-based processing underexplored. Meanwhile, LLM-based approaches for irregular time series (Kwon et al., 2025; Zhang et al., 2025) focus on purely numerical

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

observations without the textual modalities. Moreover, no prior work has examined whether and how LLMs can leverage sampling patterns in MITS as predictive signal in their own right. To this end, we develop an LLM-based framework for classifying MITS with informative sampling.

We introduce **MILM** (**M**ultimodal **I**rregular time series **L**anguage **M**odel), a framework that serializes MITS into a time-ordered sequence of time-channel-value triplets and adapts an LLM through a two-stage fine-tuning strategy. Expressing observations in the LLM's native token vocabulary enables the model to draw on its pretrained knowledge, while the global time ordering makes the causal structure of the data and sampling processes explicit to the model. In Stage 1, MILM performs value redaction fine-tuning, teaching the model to predict from sampling patterns alone rather than defaulting to the more apparent value signal. In Stage 2, it continues fine-tuning on the full data to jointly utilize observed values and the learned sampling patterns.

Empirically, MILM achieves the best average rank of 1.1 against 15 baselines across 4 EHR datasets and two metrics. Ablations show that MILM outperforms a variant that skips Stage 1 (MILM-Direct) in all 8 dataset-metric combinations, while spending the same extra epochs on full data improves over MILM-Direct in only 3 of 8. Time-ordered serialization also outperforms channel-ordered serialization in 7 of 8 combinations. Value redaction evaluation confirms that sampling patterns carry predictive signal, that Stage 1 learns to exploit them, and that Stage 2 retains most of this ability. We further introduce value pending evaluation, where some values are unavailable at prediction time, and MILM remains ahead of MILM-Direct when the affected observations are dropped, with the lead widening when their time and channel are shown. In summary, our contributions are:

- We present the first study of LLMs for MITS classification and the first investigation of whether and how LLMs can learn to exploit informative sampling patterns in this setting.
- We propose **MILM**, which represents MITS as a time-ordered sequence of time-channel-value triplets in the LLM's native token vocabulary and applies two-stage fine-tuning to first learn from sampling patterns and then jointly model sampling patterns and values.
- We present extensive experiments on 4 EHR-derived MITS datasets, where MILM achieves the best average rank against 15 baselines. Ablations together with value redaction and value pending evaluations show that it effectively learns and uses informative sampling patterns.

2 Related Work

Irregular Time Series. Irregular time series (Shukla & Marlin, 2020) arise when observations arrive at non-uniform time intervals and different variables are recorded asynchronously. One common approach is to represent each sample with aligned value, mask, and time-gap matrices, allowing sequence models to exploit observed values and missingness patterns (Che et al., 2018). Other methods such as IP-Net (Shukla & Marlin, 2019), mTAND (Shukla & Marlin, 2021), and ATENet (Lee et al., 2025b) learn interpolation representations at reference times. Another line of work explicitly models continuous-time latent dynamics with ODEs (Rubanova et al., 2019; De Brouwer et al., 2019), CDEs (Kidger et al., 2020), and SDEs (Schirmer et al., 2022; Ansari et al., 2023). Several recent approaches adapt attention (Chen et al., 2023; Zhang et al., 2023a) or recast irregular time series as sets (Horn et al., 2020), graphs (Zhang et al., 2022; 2024b; Yalavarthi et al., 2024; Luo et al., 2024), or images (Li et al., 2023). The above approaches do not consider the multimodal setting where channels may also include free text.

Multimodal Irregular Time Series. Multimodal irregular time series (MITS) (Chang et al., 2025) extend the irregular time series setting beyond purely numerical observations to include additional modalities such as text. Khadanga et al. (2019) use timestamped clinical notes with numerical time series for ICU prediction, but handle the irregular note stream with simple temporal weighting. UTDE (Zhang et al., 2023b) uses mTAND (Shukla & Marlin, 2021) to model both irregular numerical observations and timestamped clinical note embeddings. The resulting representations of the text and numerical channels are then fused with self- and cross-attention blocks for prediction. FuseMoE (Han et al., 2024) further extends multimodal fusion to fleximodal settings with additional modalities and mixture-of-experts routing. Time-IMM (Chang et al., 2025) is a recent benchmark for forecasting on irregular multimodal time series. In contrast, we study time-series-wise MITS classification where each full trajectory is itself a labeled sample, in line with prior clinically motivated multimodal

prediction settings (Zhang et al., 2023b; Khadanga et al., 2019; Han et al., 2024). To the best of our knowledge, prior work has not studied the use of LLMs for jointly processing the temporal structure, numerical values, and textual content under this setting.

LLMs for Regular Time Series. Prior LLM-for-time-series methods have largely focused on regularly sampled numerical series, either by serializing values into text for direct prompting (Xue & Salim, 2023; Gruver et al., 2023) or by aligning numerical sequences to the LLM representation space (Cao et al., 2023; Zhou et al., 2023; Jin et al., 2024; Sun et al., 2023; Liu et al., 2024; Pan et al., 2024; Liu et al., 2025b;a). There is also a line of work on multimodal settings (Belyaeva et al., 2023; Chan et al., 2024; Jia et al., 2024; Cheng et al., 2025; Wang et al., 2025; Lee et al., 2025a; Langer et al., 2025) where time series are combined with text used as context or summaries rather than as observations within the temporal sequence. Adjacent work on time-series foundation models (Das et al., 2023; Shi et al., 2024; Ansari et al., 2024; 2025) also relies on large transformer architectures, but is pretrained on time-series corpora rather than natural language.

LLMs for Irregular Time Series and Event Streams. Aside from regularly sampled time series, LLMs have also been applied to continuous-time event streams (Liu & Quan, 2024; Gupta et al., 2025; Kong et al., 2026). These works focus on temporal point process problems such as next-event time/type prediction and intensity estimation, whereas we study time-series-wise prediction from timestamped channel-value observations whose values can be numerical or textual. On the other hand, LLMs for purely numerical irregular time series (Kwon et al., 2025; Zhang et al., 2025) have also been explored. VITAL (Kwon et al., 2025) extends Time-LLM (Jin et al., 2024), a representation alignment approach for regularly sampled time series, to handle missingness. ISTS-PLM (Zhang et al., 2025) uses time, variable, and value embedders to encode irregular numerical observations before processing them with a pretrained language model. In contrast, we study LLMs for multimodal irregular time series, where observations arrive asynchronously across channels and their values may be either numerical measurements or unstructured text.

Informative Presence/Missingness/Sampling. Informative presence or missingness (Sisk et al., 2021; Tan et al., 2023) refers to the setting where the presence or absence of data is itself informative because data are often missing not at random (Little & Rubin, 2019). Informative sampling (Lin et al., 2004; Vanderschueren et al., 2023) extends this idea to time: the fact that measurements are taken at particular times can also carry signal. Prior work has examined informative sampling as a source of bias (Lin et al., 2004; Gasparini et al., 2020; Vanderschueren et al., 2023; Jeanselme et al., 2025). Related ideas appear in irregular time series models that instead treat sampling patterns as part of the predictive signal, for example through measurement intensity (Shukla & Marlin, 2019; Rubanova et al., 2019) or elapsed time and missingness indicators (Che et al., 2018). More recently, Kobayashi et al. (2025) study the impact of informative missingness on LLM predictions from static data. In contrast, we study whether LLMs can learn and use informative sampling patterns in the temporal setting of multimodal irregular time series.

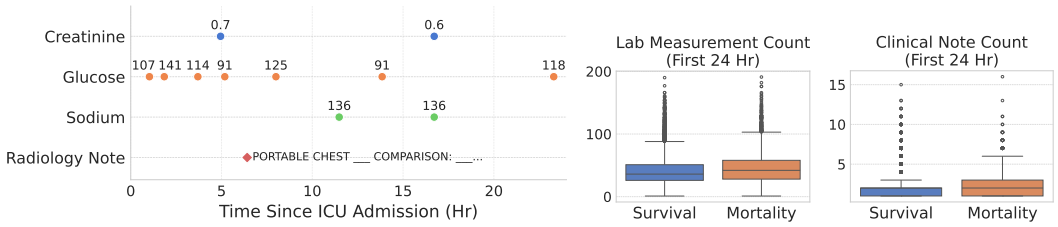
3 Problem Formulation

3.1 Multimodal Irregular Time Series

We consider multimodal irregular time series (MITS) with observations at arbitrary time points within a time range $\mathcal{I}_T = [T_{\text{start}}, T_{\text{end}}]$. We denote the set of channel types by $\mathcal{C}$. In the EHR example (Fig. 1a), $\mathcal{C}$ could contain numerical laboratory test channels and textual clinical note channels. For each channel $c \in \mathcal{C}$, we consider two interacting processes.

Data Process. Let $X_c(t) : \mathcal{I}_T \to \mathcal{X}_c$ denote the value of channel c at time t, where $\mathcal{X}_c$ is the channel value space (e.g., $\mathbb{R}$ for numerical measurements, a string space for clinical notes). This process is only observed when channel c is sampled.

Sampling Process. Let $N_c(t) : \mathcal{I}_T \to \mathbb{N}$ be a counting process recording the cumulative number of observations of channel c up to time t. It increments by 1 whenever channel c is observed: $dN_c(t) = 1$ if an observation occurs at time t, and $dN_c(t) = 0$ otherwise.



(a) (b) (c)

Figure 1: 1a: A patient's lab measurement and clinical note trajectory from the first 24 hours of an ICU stay, drawn from the MIMIC-IV-Demo database (Johnson et al., 2023a). 1b and 1c: Distributions of numbers of lab measurements and clinical notes within 24 hours grouped by outcome. Although counts are rough summaries of sampling patterns, mortality cases already exhibit more frequent observations. The full sampling patterns that encompass timings, channel identities, and orderings carry finer-grained signals that may be even more predictive of the outcome.

For each channel c, let $t_1^{(c)} < \cdots < t_{n_c}^{(c)}$ denote the times at which $dN_c(t) = 1$, i.e. the $n_c = N_c(T_{\text{end}})$ observation times of channel c. The per-channel observation sequence is $\mathbf{s}_c = ((t_k^{(c)}, x_k^{(c)}))_{k=1}^{n_c}$, where $x_k^{(c)} = X_c(t_k^{(c)})$ is the value or content. The full MITS is the collection $\{\mathbf{s}_c | c \in \mathcal{C}\}$.

3.2 Multimodal Irregular Time Series Classification

Let $Y \in \{1, \ldots, K\}$ denote the label of a MITS sample. We observe a dataset $\mathcal{D} = \{\mathcal{S}_i\}_{i=1}^M$ consisting of M i.i.d. realizations of the random observed data object $\mathcal{S} = (\{\mathbf{s}_c | c \in \mathcal{C}\}, Y)$. The goal is to learn a MITS classifier f that minimizes the expected classification loss $\mathbb{E}_{\mathcal{S}}[l(f(\{\mathbf{s}_c | c \in \mathcal{C}\}), Y)]$. Each $\mathbf{s}_c$ encodes two sources of information: (i) the observed values $\{x_k^{(c)}\}$ that directly reflect the data process and (ii) the sampling patterns $\{t_k^{(c)}\}$ and n_c, which are also informative about Y because sampling may depend on both the observed and unobserved parts of the data process. For example, a patient with a critically abnormal lab result or a worsening clinical state not yet documented in the system may be assigned more follow-up tests. Indeed, in the MIMIC-IV EHR database (Johnson et al., 2023c; 2024; 2023b), patient stays resulting in in-hospital mortality exhibit more lab measurements and clinical notes than surviving stays (Fig. 1b and 1c), though the total count is merely a coarse summary. An effective MITS classifier should leverage both observed values and sampling patterns.

4 Method

We introduce **MILM**: **M**ultimodal **I**rregular time series **L**anguage **M**odel (Fig. 2), a framework that serializes MITS into time-ordered time-channel-value triplets and employs a two-stage fine-tuning strategy to adapt an LLM to exploit both sampling patterns and values for MITS classification.

4.1 MITS Representation

We design a MITS representation that enables LLMs to process numerical and textual observations. Prior work on LLMs for irregular time series (Kwon et al., 2025; Zhang et al., 2025) views time series observations as numerical objects and transforms them into a language model's token embedding space, using representations foreign to the LLM's pretraining. This limits the model's ability to draw on pretrained knowledge. We instead represent MITS directly in the LLM's native token vocabulary, enabling joint processing of numerical and text modalities without representational alignment. Expressing timings and channel identities in natural language further grounds each value in semantic context rather than leaving it as a bare number or string. We explain the process below:

Time-Ordered Serialization. As shown in Fig. 2, a MITS $\{\mathbf{s}_c | c \in \mathcal{C}\}$ is a sparse 2-D object indexed by channel and time. We apply time-ordered serialization, augmenting each observation $(t_k^{(c)}, x_k^{(c)}) \in \mathbf{s}_c$ with its channel label c and globally sorting by time, breaking ties by predefined

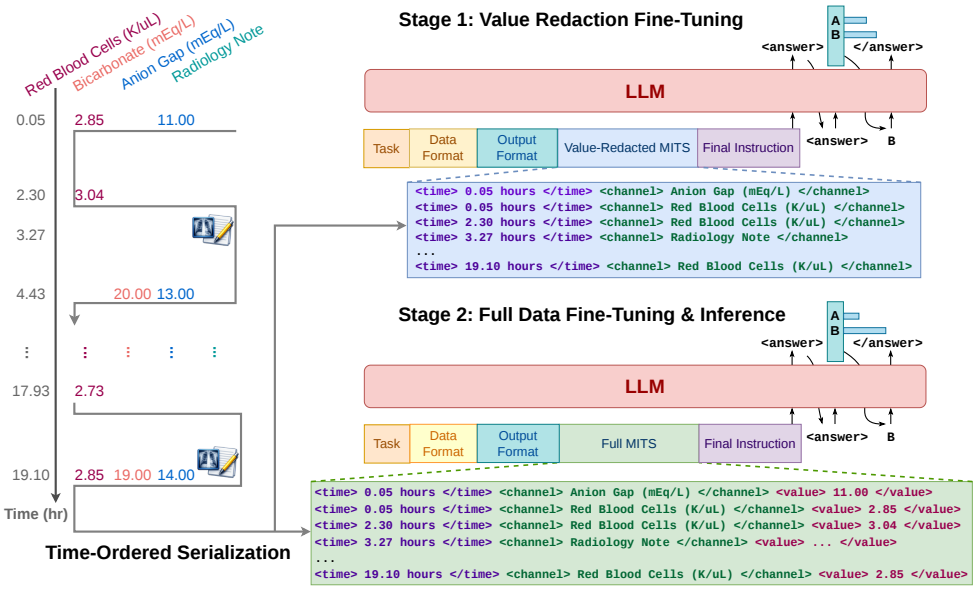


Figure 2: An illustration of MILM. MILM serializes the MITS to a time-ordered XML representation and fine-tunes the LLM in two stages. Value-redaction fine-tuning first teaches the model to predict from sampling patterns alone, while full-data fine-tuning then incorporates value information on top of the developed sampling-pattern knowledge. During inference, MILM uses the Stage 2 prompt format and extracts the answer-token logits as prediction scores.

channel order, to obtain a flattened sequence $((t_i, c_i, x_i))_{i=1}^{n}$, where t_i, c_i, and x_i are the time, channel, and value of the i-th observation, $n = \sum_{c \in \mathcal{C}} n_c$, and $t_i \leq t_j$ for all $i < j$. Preserving global temporal order ensures that the causal structure of the sampling and data processes is directly reflected in the sequence and explicit to the model.

XML-Formatted Triplet Representation. Each observation (t_i, c_i, x_i) in the serialized sequence has three semantically distinct components: when it occurred, which channel was observed, and what value was recorded. We represent each observation as a triplet in Extensible Markup Language (XML) format with tags `<time>`, `<channel>`, and `<value>`, explicitly encoding this three-part structure. Consecutive observations are separated by a newline. For example, the following snippet specifies a MITS with 3 observations from the glucose, anion gap, and radiology note channels:

```
<time> 0.88 hours </time> <channel> Glucose (mg/dL) </channel> <value> 170.00 </value>
<time> 3.80 hours </time> <channel> Anion Gap (mEq/L) </channel> <value> 11.00 </value>
<time> 4.00 hours </time> <channel> Radiology Note </channel> <value> Portable chest ... </value>
```

4.2 LLM For MITS Classification

We describe how we use LLMs for MITS classification, covering prompt and target construction for both inference and supervised fine-tuning (SFT), and prediction score extraction at inference time.

Prompt and Target Construction. To condition the LLM for the prediction task, the prompt consists of three specification blocks followed by the serialized MITS and a final instruction. The **task specification** describes the data domain and the prediction target, e.g., "timestamped observation from an ICU stay" and "predict whether the patient will survive their hospital admission". The **data format specification** explains the triplet XML format and the semantics of each tag (e.g., the `<time>` tag records hours since the start of the ICU stay). The **output format specification** maps each class label to an alphabet letter and instructs the model to place its prediction in an `<answer>` tag (e.g., `<answer> B </answer>`). See Appendix A for the full prompt templates. For SFT, the training target is the ground-truth class letter and the enclosing `<answer>` tags. The model is trained via next-token prediction loss on the full target sequence.

Prediction Score Extraction. Following Robinson & Wingate (2023), at inference we extract prediction scores from the logits at the position following `<answer>` and apply softmax over the class letter tokens to obtain a probability distribution. We prefer this over verbalizing class probabilities, which would require numerical confidence targets during SFT that are typically unavailable.

4.3 Two-Stage Fine-Tuning

The signal in sampling patterns is subtler than that in values, particularly when text channels are present and value tokens constitute a large portion of the sequence. To avoid bypassing sampling patterns in favor of the more apparent value signal, we propose a two-stage fine-tuning strategy that forces sampling-pattern representations to develop before value information is introduced.

Stage 1: Value Redaction Fine-Tuning. In the first stage, we fine-tune the LLM on MITS with all values redacted. Each observation is presented as a time-channel pair, as in the following example:

```
<time> 0.88 hours </time> <channel> Glucose (mg/dL) </channel>
<time> 3.80 hours </time> <channel> Anion Gap (mEq/L) </channel>
<time> 4.00 hours </time> <channel> Radiology Note </channel>
```

The data format specification in the prompt is adjusted to reflect the redacted format. An additional note clarifies that values were recorded but withheld and instructs the model to rely on timing and channel patterns for prediction. Stage 1 trains the model to develop predictive representations of the sampling processes $\{N_c(t)|c \in \mathcal{C}\}$ through the timings and channel identities of observations.

Stage 2: Full Data Fine-Tuning. In the second stage, we continue fine-tuning from the Stage 1 checkpoint on the full MITS data, using the same representation and prompt construction described in Sections 4.1 and 4.2. Building on the sampling-pattern representations developed in Stage 1, Stage 2 trains the model to jointly exploit both the data processes $\{X_c(t)|c \in \mathcal{C}\}$ and the sampling processes $\{N_c(t)|c \in \mathcal{C}\}$. The Stage 2 model is used for the final prediction.

This two-stage scheme is enabled by the text-based representation: removing the `<value>` tokens preserves the presence of each observation while unambiguously signaling that its value is withheld, allowing the same model architecture to process both redacted and full MITS without modification.

5 Experiments

5.1 Setup

Data. We conduct experiments on datasets extracted from two major public EHR databases: MIMIC-IV (Johnson et al., 2023c; 2024; 2023b) and eICU Collaborative Research Database (Pollard et al., 2018; 2019). Following Jeanselme et al. (2025), we extract the laboratory tests as the numerical channels due to their inherently irregular sampling nature. In addition, we extract the clinical notes as the text channel. On each database, we develop two datasets corresponding to the in-hospital mortality (IHM) and length-of-stay (LOS) prediction tasks (Harutyunyan et al., 2019; Sheikhalishahi et al., 2020) given the first 24 hours of laboratory tests and clinical notes of the ICU stay. We describe the datasets in detail in Appendix B.

Baselines. We select 3 classes of irregular time series model baselines: (i) Non-LLM-based: GRU-D (Che et al., 2018), mTAND (Shukla & Marlin, 2021), KEDGN (Luo et al., 2024), and VITAL-stats (Kwon et al., 2025), (ii) LLM-based: VITAL-LLM (Kwon et al., 2025) and ISTS-PLM (Zhang et al., 2025), and (iii) Multimodal: UTDE (Zhang et al., 2023b) and FuseMoE (Han et al., 2024). Since the datasets belong to the biomedical domain, we use BioBERT (Lee et al., 2020) as the note embedding model. The unimodal baselines do not use the text channel by default. For those models, we implement a "+ Note" version that uses the timestamp-to-text fusion (TTF) module proposed by Chang et al. (2025) to learn the text channel representation to be concatenated with the numerical channel representation before the prediction head.

MILM Setup. We fine-tune Qwen3-4B-Instruct-2507 (Yang et al., 2025) using QLoRA (Dettmers et al., 2023) with rank $r = 16$, $\alpha = 16$, and dropout 0.05 applied to the query, key, value, and output projection matrices of the LLM. We use a learning rate of 10^{-4} with a cosine schedule and warmup

Table 1: Results on the in-hospital mortality (IHM) and length-of-stay (LOS) prediction datasets extracted from MIMIC-IV and eICU. Best results are **boldfaced** and second-best results are underlined.

Method	MIMIC-IV-IHM		MIMIC-IV-LOS		eICU-IHM		eICU-LOS		Avg. Rank
	AU-ROC	AP	AU-ROC	AP	AU-ROC	AP	AU-ROC	AP	
GRU-D	75.91±1.03	54.08±1.02	70.08±0.83	46.89±1.17	66.04±0.99	34.98±3.05	60.48±2.14	51.13±3.09	11.0
+ Note	74.52±1.71	52.28±4.13	70.93±2.54	47.84±2.45	66.85±1.21	35.97±1.18	57.40±3.51	45.84±4.02	12.5
mTAND	77.38±0.90	56.52±1.55	70.98±0.96	48.99±1.68	67.13±1.35	36.42±3.11	58.10±1.77	50.36±1.29	9.3
+ Note	77.90±1.16	58.57±0.68	74.30±0.82	54.26±1.76	66.99±1.33	37.11±2.52	64.79±1.03	52.20±0.64	5.9
KEDGN	79.59±1.21	60.77±2.23	71.86±1.82	48.44±2.26	71.60±1.55	41.24±1.30	65.44±1.65	55.49±3.15	4.4
+ Note	72.04±4.09	46.79±4.23	62.95±5.18	40.55±4.51	70.50±1.13	39.37±0.89	63.18±1.90	51.71±3.05	10.4
UTDE	75.23±3.10	56.14±3.72	75.68±2.78	56.50±3.53	**72.98±0.92**	41.58±1.16	63.96±3.05	50.35±4.23	5.4
FuseMoE	76.21±1.45	56.27±0.78	76.13±0.24	57.03±1.62	70.57±0.75	40.43±1.27	65.37±2.15	52.48±3.29	4.5
VITAL-stats	76.99±0.66	54.06±2.06	73.62±0.96	52.02±2.89	70.51±1.48	39.36±2.68	60.01±2.42	48.12±1.56	8.9
+ Note	77.80±0.68	54.32±1.67	73.99±0.98	53.83±3.61	69.77±1.05	38.54±2.01	59.83±1.07	48.67±1.42	8.3
VITAL-LLM	66.43±2.23	39.61±2.45	67.60±1.29	45.94±2.75	61.92±2.72	28.68±2.44	60.14±2.04	50.14±2.51	13.9
+ Note	67.02±2.34	39.22±3.20	67.97±0.99	47.49±1.52	61.19±2.51	28.09±2.53	59.91±2.13	49.63±1.64	14.1
ISTS-PLM	80.49±0.79	60.10±1.14	73.92±0.87	54.04±1.77	68.93±0.72	36.32±1.18	62.66±1.33	52.79±1.44	5.8
+ Note	80.07±0.52	59.06±0.62	73.68±0.61	52.09±1.13	69.02±0.65	36.12±0.32	62.51±1.39	52.82±1.38	6.4
Qwen3-4B	69.82±0.00	48.36±0.00	67.98±0.00	44.82±0.00	64.53±0.00	29.43±0.00	55.71±0.00	45.58±0.00	14.4
MILM	**81.30±0.70**	**63.86±1.47**	**80.02±0.55**	**59.73±1.21**	72.47±1.49	**42.27±2.10**	**66.05±1.79**	**56.22±1.20**	**1.1**

ratio 0.05. Stage 1 of MILM trains for 8 epochs. Stage 2 initializes from the best Stage 1 checkpoint, and the number of training epochs is dataset-dependent. At inference, we use SGLang (Zheng et al., 2024) to serve the fine-tuned model in bfloat16 precision with logit-based prediction score extraction. We also evaluate the off-the-shelf Qwen3-4B-Instruct-2507 (hereafter Qwen3-4B) as a zero-shot baseline. See Appendix C for further details regarding implementation and hyper-parameters.

Evaluation. Each dataset is split into $70\%/15\%/15\%$ for train/validation/test. We use Area Under the Receiver Operating Characteristic Curve (AU-ROC) and Average Precision (AP) to evaluate the quality of the classification scores and report the average and standard deviation over 5 runs.

5.2 Main Results

Table 1 summarizes performance across all four datasets. MILM achieves the best average rank of 1.1 across all eight metric-dataset combinations. Among the baselines, KEDGN (avg. rank 4.4), FuseMoE (avg. rank 4.5), and UTDE (avg. rank 5.4) perform best, and all three natively incorporate language semantics. KEDGN encodes the semantic representation of each channel with a pretrained language model, while FuseMoE and UTDE fuse information between the note embedding and numerical observation sequences with mixture-of-experts and cross-attention. In contrast, simply augmenting unimodal models with a separately computed text channel representation via concatenation ("+ Note") often fails to help and sometimes hurts performance, suggesting that the integration strategy matters as much as having the text channel at all. MILM takes a complementary approach by interleaving numerical and text observations while preserving causal structure, allowing the LLM to jointly attend to both modalities and achieve the best overall performance.

Comparing MILM against the zero-shot Qwen3-4B baseline illuminates the contribution of fine-tuning. Qwen3-4B achieves above-random performance (e.g., AU-ROC of 69.82 on MIMIC-IV-IHM), confirming that our constructed prompt conditions the LLM to effectively leverage its pretrained knowledge for the task even without fine-tuning. Yet it ranks last overall (avg. rank 14.4), indicating that domain-specific MITS patterns require further adaptation. MILM substantially improves upon this zero-shot baseline. Among the LLM-based irregular time series baselines, VITAL-LLM performs the weakest overall (avg. rank 13.9), even trailing Qwen3-4B on several datasets. Both VITAL-LLM and ISTS-PLM require aligning numerical observations to the LLM's token embedding space via text prototype reprogramming in VITAL-LLM (Kwon et al., 2025), and via learnable time, variable, and value embedders in ISTS-PLM (Zhang et al., 2025). MILM sidesteps this alignment problem entirely by adopting a text-based representation, preserving the LLM's pretrained token embeddings and the knowledge encoded within them, and outperforms both baselines across all four datasets.

Table 2: Comparison of MILM-Direct, MILM-Direct-Eq, and MILM.

	MIMIC-IV-IHM		MIMIC-IV-LOS		eICU-IHM		eICU-LOS	
Method	AU-ROC	AP	AU-ROC	AP	AU-ROC	AP	AU-ROC	AP
MILM-Direct	80.82±0.63	63.69±1.29	78.56±1.26	58.85±1.37	71.08±0.69	41.77±1.72	64.97±1.42	55.76±1.08
MILM-Direct-Eq	80.38±1.00	**63.91±0.83**	78.17±2.31	59.54±2.16	71.94±1.40	41.48±2.17	64.51±1.85	55.52±1.89
MILM	**81.30±0.70**	63.86±1.47	**80.02±0.55**	**59.73±1.21**	**72.47±1.49**	**42.27±2.10**	**66.05±1.79**	**56.22±1.20**

Table 3: Comparison of MILM-Direct with channel-ordered and time-ordered serialization.

	MIMIC-IV-IHM		MIMIC-IV-LOS		eICU-IHM		eICU-LOS	
Method	AU-ROC	AP	AU-ROC	AP	AU-ROC	AP	AU-ROC	AP
Channel-Ordered	80.56±0.64	60.73±0.96	76.49±1.14	**59.36±1.39**	70.71±0.98	41.08±1.47	64.30±1.14	54.73±1.00
Time-Ordered	**80.82±0.63**	**63.69±1.29**	**78.56±1.26**	58.85±1.37	**71.08±0.69**	**41.77±1.72**	**64.97±1.42**	**55.76±1.08**

5.3 Ablation Study

We conduct two ablation studies. The first studies the effect of value redaction fine-tuning in MILM. We consider two single-stage variants: MILM-Direct and MILM-Direct-Eq. MILM-Direct skips Stage 1 and directly performs full data fine-tuning for the same number of epochs as MILM's Stage 2. MILM-Direct-Eq also only performs full data fine-tuning, but for as many epochs as MILM's two stages combined. We present the results in Table 2. MILM outperforms MILM-Direct in all 8 dataset-metric combinations and outperforms MILM-Direct-Eq in 7 of 8. Compared to MILM-Direct, MILM-Direct-Eq spends the additional epochs on full data fine-tuning and improves in only 3 of 8 combinations. Reallocating the same epoch budget to value redaction fine-tuning improves over MILM-Direct in all 8. Therefore, first learning from sampling patterns alone provides a better foundation for full data fine-tuning than simply extending it.

The second ablation studies time-ordered serialization, which orders observations chronologically and breaks ties with a predefined channel order. An alternative is channel-ordered serialization, which groups observations of the same channel together and sorts them by time within each channel but not globally. We compare the two under MILM-Direct, which isolates the serialization choice from the two-stage scheme, and present the results in Table 3. Time-ordered serialization outperforms channel-ordered serialization in 7 of 8 dataset-metric combinations, suggesting that keeping the causal structure of the sampling and data processes apparent in the serialized sequence is helpful.

5.4 Value Redaction Evaluation

We conduct value redaction evaluation to understand how well each model exploits sampling patterns. We test off-the-shelf Qwen3-4B, MILM-Direct, and the Stage 1 and Stage 2 checkpoints of MILM with all observed values removed, keeping only the time and channel information. The results are shown in Fig. 3. MILM-Direct does not improve over Qwen3-4B on MIMIC-IV, but does on eICU, where Qwen3-4B performs worse than random. MILM (Stage 1) shows substantial improvement over both Qwen3-4B and MILM-Direct, demonstrating that sampling patterns are informative and can be learned through value redaction fine-tuning. MILM (Stage 2) typically experiences a small degree of forgetting (e.g., a maximum decrease of 2.89 AU-ROC on MIMIC-IV-LOS), with the exception of MIMIC-IV-IHM, where performance slightly increases. Nevertheless, MILM (Stage 2) still predicts more accurately from sampling patterns alone than both Qwen3-4B and MILM-Direct, showing that it retains most sampling-pattern knowledge after full-data fine-tuning.

5.5 Value Pending Evaluation

We introduce value pending evaluation, motivated by a practical constraint: some lab measurements or notes are taken before the prediction time, but their results are not yet available in the database when the model is queried (Otles et al., 2021). For these observations, the time and channel are known but the values are not. We consider two countermeasures, dropping these observations entirely or showing their presence without revealing the values, as in the following snippet:

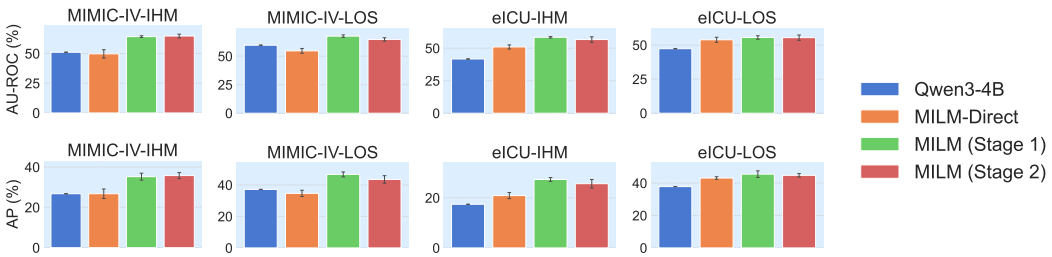


Figure 3: Value redaction evaluation across datasets. All models are tested with observed values removed and only time and channel information retained. MILM (Stage 1) substantially outperforms both Qwen3-4B and MILM-Direct, while MILM (Stage 2) retains most of the performance.

Table 4: Value pending evaluation results of MILM-Direct and MILM under the drop observation and show presence countermeasures. Best results are **boldfaced**.

(a) Drop Observation

	MIMIC-IV-IHM		MIMIC-IV-LOS	
Method	AU-ROC	AP	AU-ROC	AP
MILM-Direct	79.04±0.91	59.23±1.63	77.36±1.42	57.71±2.13
MILM	**79.49**±0.76	**60.02**±1.57	**78.65**±0.41	**58.02**±0.68

(b) Show Presence

	MIMIC-IV-IHM		MIMIC-IV-LOS	
Method	AU-ROC	AP	AU-ROC	AP
MILM-Direct	79.21±0.65	58.14±1.37	76.47±1.40	56.19±2.36
MILM	**80.34**±0.64	**61.23**±1.62	**78.14**±0.47	**57.11**±0.96

```
<time> 19.10 hours </time> <channel> Anion Gap (mEq/L) </channel> <value> 14.00 </value>
<time> 19.10 hours </time> <channel> Bicarbonate (mEq/L) </channel>
<time> 19.10 hours </time> <channel> Red Blood Cells (K/uL) </channel> <value> 2.85 </value>
```

Showing presence conveys extra sampling information but exposes the model to a mixed format of pairs and triplets it never saw in training.

In MIMIC-IV, each lab event and clinical note has two associated timestamps: `charttime` and `storetime`. `charttime` is the clinically relevant event time used for MITS construction, and `storetime` is when the result became available in the database. We withhold the values of observations whose `storetime` exceeds the prediction time of 24 hours since ICU admission. eICU does not have an equivalent pair of time columns, so we focus this evaluation on MIMIC-IV. See Appendix E for dataset statistics and full results including baselines under drop observation.

Table 4 shows that MILM outperforms MILM-Direct in all 8 dataset-metric combinations across both countermeasures. Under drop observation, MILM maintains the lead it holds under standard evaluation, indicating better use of the remaining observations. Under show presence, the lead widens in all 4 combinations. Stage 1 is trained on time-channel pairs and Stage 2 on full triplets, so MILM handles the mixed format better despite never being trained on the two together. Value-pending observations tend to fall at the end of the timeline and are often critical for predicting future outcomes (Choi et al., 2016), so we encourage follow-up work to adopt this evaluation setting.

6 Conclusion

We introduce MILM, the first LLM-based framework for classifying multimodal irregular time series (MITS) with informative sampling. MILM serializes observations into a time-ordered sequence of time-channel-value triplets and applies a two-stage fine-tuning strategy. Stage 1 teaches the LLM to predict solely from sampling patterns, while Stage 2 incorporates value information on top of the learned sampling patterns. MILM achieves the best average rank of 1.1 against 15 baselines on 4 EHR datasets. Ablations confirm the contribution of value redaction and the advantage of time-ordered over channel-ordered serialization. Value redaction evaluation further confirms that sampling patterns are predictive and that two-stage fine-tuning effectively learns and retains this predictive capability. We also introduce a value pending evaluation, where the values of some observations are unavailable at prediction time. MILM stays ahead of single-stage fine-tuning under both countermeasures, and its lead widens when the time and channel of pending observations are shown. Future work includes extending MILM to additional modalities and studying robustness under sampling distribution shift, such as deployment across hospitals with different sampling policies.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, and assist with translation. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures, create or edit software code, draft parts of this paper, edit this paper to improve readability, and identify relevant literature.

We have reviewed all AI-assisted work. AI-generated code was read, tested, and verified by the authors. Every reported number comes from a run executed and inspected by the authors. AI-assisted tables and figures were cross-checked against the underlying experimental outputs, AI-suggested references were manually verified, and all AI-assisted text was reviewed for factual accuracy. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses MIMIC-IV (Johnson et al., 2023c; 2024), MIMIC-IV-Note (Johnson et al., 2023b), and the eICU Collaborative Research Database (Pollard et al., 2018; 2019), which are de-identified and distributed through PhysioNet under the PhysioNet Credentialed Health Data License 1.5.0. All authors who accessed the data completed the required training and are covered by the corresponding data use agreements, and we made no attempt to re-identify individuals or link records to external sources. To comply with these agreements, all models were run on local compute. We release code but no data. The datasets remain available only through PhysioNet's credentialing process. Licenses for all datasets and pretrained models are listed in Appendix H.

REPRODUCIBILITY STATEMENT

Dataset extraction and preprocessing for all four datasets are described in Appendix B. Appendix C reports the configurations and hyper-parameters for all methods, and Appendix A gives the full prompt templates for each MITS representation. MIMIC-IV and eICU require credentialed access through PhysioNet and cannot be redistributed, so we make our preprocessing, training, and evaluation code available at `https://anonymous.4open.science/r/milm-7750`.

REFERENCES

Abdul Fatir Ansari, Alvin Heng, Andre Lim, and Harold Soh. Neural continuous-discrete state space models for irregularly-sampled time series. In *International Conference on Machine Learning*, pp. 926–951. PMLR, 2023.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Anastasiya Belyaeva, Justin Cosentino, Farhad Hormozdiari, Krish Eswaran, Shravya Shetty, Greg Corrado, Andrew Carroll, Cory Y McLean, and Nicholas A Furlotte. Multimodal llms for health grounded in individual-specific data. In *Workshop on Machine Learning for Multimodal Healthcare Data*, pp. 86–102. Springer, 2023.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

Nimeesha Chan, Felix Parker, William Bennett, Tianyi Wu, Mung Yao Jia, James Fackler, and Kimia Ghobadi. Medtsllm: Leveraging llms for multimodal medical time series analysis. *arXiv preprint arXiv:2408.07773*, 2024.

Ching Chang, Jeehyun Hwang, Yidan Shi, Haixin Wang, Wei Wang, Wen-Chih Peng, and Tien-Fu Chen. Time-IMM: A dataset and benchmark for irregular multimodal multivariate time series. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=yeqrrn51TL.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36:47143–47175, 2023.

Mingyue Cheng, Yiheng Chen, Qi Liu, Zhiding Liu, Yucong Luo, and Enhong Chen. Instructime: Advancing time series classification with multimodal language modeling. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pp. 792–800, 2025.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alessandro Gasparini, Keith R Abrams, Jessica K Barrett, Rupert W Major, Michael J Sweeting, Nigel J Brunskill, and Michael J Crowther. Mixed-effects models for health care longitudinal data with an informative visiting process: A monte carlo simulation study. *Statistica Neerlandica*, 74 (1):5–23, 2020.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

Shubham Gupta, Thibaut Durand, Graham W. Taylor, and Lilian Bialokozowicz. LAST SToP for modeling asynchronous time series. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21297–21321. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gupta25a.html.

Xing Han, Huy Nguyen, Carl Harris, Nhat Ho, and Suchi Saria. Fusemoe: Mixture-of-experts transformers for fleximodal fusion. *Advances in Neural Information Processing Systems*, 37: 67850–67900, 2024.

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

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *International Conference on Machine Learning*, pp. 4353–4363. PMLR, 2020.

Vincent Jeanselme, Glen Martin, Matthew Sperrin, Niels Peek, Brian Tom, and Jessica Barrett. Prediction of survival outcomes under clinical presence shift: A joint neural network architecture. *arXiv preprint arXiv:2508.05472*, 2025.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 23343–23351, 2024.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV Clinical Database Demo. *PhysioNet*, January 2023a. doi: 10.13026/dp1f-ex47. URL https://doi.org/10.13026/dp1f-ex47. Version 2.2.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet*, January 2023b. doi: 10.13026/1n74-ne17. URL https://doi.org/10.13026/1n74-ne17. Version 2.2.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV. *PhysioNet*, October 2024. doi: 10.13026/kpb9-mt58. URL https://doi.org/10.13026/kpb9-mt58. Version 3.1.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023c.

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. *arXiv preprint arXiv:1907.05321*, 2019.

Swaraj Khadanga, Karan Aggarwal, Shafiq Joty, and Jaideep Srivastava. Using clinical notes with time series data for icu management. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6432–6437, 2019.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Yuta Kobayashi, Vincent Jeanselme, and Shalmali Joshi. Mind the data gap: Missingness still shapes large language model prognoses. *arXiv preprint arXiv:2512.00479*, 2025.

Quyu Kong, Yixuan Zhang, Yang Liu, Panrong Tong, Enqi Liu, and Feng Zhou. Byte-token enhanced language models for temporal point processes analysis. In *Proceedings of the ACM Web Conference 2026*, pp. 7013–7023, 2026.

Ross Koval, Nicholas Andrews, and Xifeng Yan. Multimodal language models for financial forecasting from interleaved sequences of text and time series. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 987–1001, 2025.

Jeong Eul Kwon, Joo Heung Yoon, and Hyo Kyung Lee. Mind the missing: Variable-aware representation learning for irregular ehr time series using large language models. *arXiv preprint arXiv:2509.22121*, 2025.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A Xu, Winnie Chow, Martin Maritsch, Robert Jakob, Ning Wang, Juncheng Liu, Aradhana Verma, et al. Opentslm: Time-series language models for reasoning over multivariate medical text-and time-series data. *arXiv preprint arXiv:2510.02410*, 2025.

Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. Timecap: Learning to contextualize, augment, and predict time series events with large language model agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18082–18090, 2025a.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Sangho Lee, Kyeongseo Min, Youngdoo Son, and Hyungrok Do. Adaptive time encoding for irregular multivariate time-series classification. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36:49187–49204, 2023.

Haiqun Lin, Daniel O Scharfstein, and Robert A Rosenheck. Analysis of longitudinal data with irregular, outcome-dependent follow-up. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(3):791–813, 2004.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2019.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18780–18788, 2025a.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025b.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024.

Zefang Liu and Yinzhu Quan. Tpp-llm: Modeling temporal point processes by efficiently fine-tuning large language models. *arXiv preprint arXiv:2410.02062*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yicheng Luo, Zhen Liu, Linghao Wang, Junhao Zheng, Binquan Wu, and Qianli Ma. Knowledge-empowered dynamic graph network for irregularly sampled medical time series. *Advances in neural information processing systems*, 37:67172–67199, 2024.

Erkin Otles, Jeeheh Oh, Benjamin Li, Michelle Bochinski, Hyeon Joo, Justin Ortwine, Erica Shenoy, Laraine Washer, Vincent B Young, Krishna Rao, et al. Mind the performance gap: examining dataset shift during prospective validation. In *Machine Learning for Healthcare Conference*, pp. 506–534. PMLR, 2021.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. S^2ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Tom Pollard, Alistair Johnson, Jesse Raffa, Leo Anthony Celi, Omar Badawi, and Roger Mark. eICU Collaborative Research Database. *PhysioNet*, April 2019. doi: 10.13026/C2WM1R. URL https://doi.org/10.13026/C2WM1R. Version 2.0.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):180178, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Joshua Robinson and David Wingate. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=yKbprarjc5B.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International conference on machine learning*, pp. 19388–19405. PMLR, 2022.

Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cían Hughes, Charles Lau, et al. Medgemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. Benchmarking machine learning models on multi-centre eicu critical care dataset. *Plos one*, 15(7):e0235424, 2020.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

Satya Narayan Shukla and Benjamin Marlin. Interpolation-prediction networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1efr3C9Ym.

Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=4c0J6lwQ4_.

Satya Narayan Shukla and Benjamin M Marlin. A survey on principles, models and methods for learning from irregularly sampled time series. *arXiv preprint arXiv:2012.00168*, 2020.

Rose Sisk, Lijing Lin, Matthew Sperrin, Jessica K Barrett, Brian Tom, Karla Diaz-Ordaz, Niels Peek, and Glen P Martin. Informative presence and observation in routine health data: a review of methodology for clinical risk prediction. *Journal of the American Medical Informatics Association*, 28(1):155–166, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

Amelia LM Tan, Emily J Getzen, Meghan R Hutch, Zachary H Strasser, Alba Gutiérrez-Sacristán, Trang T Le, Arianna Dagliati, Michele Morris, David A Hanauer, Bertrand Moal, et al. Informative missingness: What can we learn from patterns in missing laboratory data in the electronic health record? *Journal of biomedical informatics*, 139:104306, 2023.

Toon Vanderschueren, Alicia Curth, Wouter Verbeke, and Mihaela Van Der Schaar. Accounting for informative sampling when learning to forecast treatment outcomes over time. In *International Conference on Machine Learning*, pp. 34855–34874. PMLR, 2023.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025.

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. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2023.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. Grafiti: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16255–16263, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3273–3285, 2023a.

Lingzhe Zhang, Tong Jia, Mengxi Jia, Yifan Wu, Aiwei Liu, Yong Yang, Zhonghai Wu, Xuming Hu, Philip S Yu, and Ying Li. A survey of aiops for failure management in the era of large language models. *arXiv preprint arXiv:2406.11213*, 2024a.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *Forty-first International Conference on Machine Learning*, 2024b.

Weijia Zhang, Chenlong Yin, Hao Liu, and Hui Xiong. Unleashing the power of pre-trained language models for irregularly sampled time series. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3831–3842, 2025.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Kwm8I7dU-l5.

Xinlu Zhang, Shiyang Li, Zhiyu Chen, Xifeng Yan, and Linda Ruth Petzold. Improving medical predictions by irregular multimodal electronic health records modeling. In *International conference on machine learning*, pp. 41300–41313. PMLR, 2023b.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A Prompt Templates

We present the prompt templates for the in-hospital mortality (IHM) and length-of-stay (LOS) prediction tasks. The templates are organized into four groups based on the MITS representation used. The first group uses the full MITS representation, corresponding to Stage 2 of MILM and inference. The second group uses the value-redacted MITS representation, corresponding to Stage 1 of MILM. The third group corresponds to value pending evaluation with show presence, where some observations have been measured, but their values are unavailable at prediction time and are shown as time-channel pairs. The fourth group corresponds to the channel-ordered serialization ablation, where observations are grouped by channel instead of being ordered globally by time.

A.1 Prompt Templates for Full MITS

Full MITS Prompt Template: In-Hospital Mortality Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Analyze timestamped observations from an ICU stay and predict whether the patient
will survive their hospital admission.

DATA FORMAT: Each observation is a triplet:
   <time> hours_since_start </time> <channel> channel_name </channel> <value> value
</value>

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) SURVIVAL
(B) MORTALITY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (SURVIVAL): <answer> A </answer>
Example 2 (MORTALITY): <answer> B </answer>

[OBSERVATIONS]
...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will experience in-hospital
mortality.
Follow the output format exactly.  Start directly with <answer>.
```

Full MITS Prompt Template: Length-of-Stay Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Given timestamped observations from the first 24-hour window of an ICU stay, predict
whether the patient will ultimately have a short ICU stay (<96 hours) and survive.

DATA FORMAT: Each observation is a triplet:
   <time> hours_since_start </time> <channel> channel_name </channel> <value> value
</value>

CLASS DEFINITION:
- SHORT_STAY: ICU stay < 96 hours and survived.
- LONG_STAY: ICU stay >= 96 hours or death.
```

Prompt Template (continued)

```
OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) LONG_STAY
(B) SHORT_STAY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (LONG_STAY): <answer> A </answer>
Example 2 (SHORT_STAY): <answer> B </answer>

[OBSERVATIONS]
...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will have a short ICU stay
and survive.
Follow the output format exactly.  Start directly with <answer>.
```

A.2 Prompt Templates for Value-Redacted MITS

Value-Redacted MITS Prompt Template: In-Hospital Mortality Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Analyze timestamped observations from an ICU stay and predict whether the patient
will survive their hospital admission.

DATA FORMAT: Each observation is a pair:
<time> hours_since_start </time> <channel> channel_name </channel>

IMPORTANT:
- For each observation, the value/content was measured and recorded, but is not provided
here.
- When making the prediction, use the information conveyed by the timings and channel
types themselves | what types of laboratory measurements were ordered and what notes
were recorded, when, in what sequence, and how frequently | and what these patterns might
signal about the patient's clinical trajectory and mortality risk.

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) SURVIVAL
(B) MORTALITY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (SURVIVAL): <answer> A </answer>
Example 2 (MORTALITY): <answer> B </answer>
```

Prompt Template (continued)

```
[OBSERVATIONS]
...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will experience in-hospital
mortality.
Follow the output format exactly.  Start directly with <answer>.
```

Value-Redacted MITS Prompt Template: Length-of-Stay Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Given timestamped observations from the first 24-hour window of an ICU stay, predict
whether the patient will ultimately have a short ICU stay (<96 hours) and survive.

DATA FORMAT: Each observation is a pair:
<time> hours_since_start </time> <channel> channel_name </channel>

CLASS DEFINITION:
- SHORT_STAY: ICU stay < 96 hours and survived.
- LONG_STAY: ICU stay >= 96 hours or death.

IMPORTANT:
- For each observation, the value/content was measured and recorded, but is not provided
here.
- When making the prediction, use the information conveyed by the timings and channel
types themselves - what types of laboratory measurements were ordered and what notes
were recorded, when, in what sequence, and how frequently - and what these patterns might
signal about whether the patient's ICU course is likely to stabilize quickly (short stay
and survive) versus become prolonged or deteriorate (ICU stay >= 96 hours or death).

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) LONG_STAY
(B) SHORT_STAY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (LONG_STAY): <answer> A </answer>
Example 2 (SHORT_STAY): <answer> B </answer>

[OBSERVATIONS]
...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will have a short ICU stay
and survive.
Follow the output format exactly.  Start directly with <answer>.
```

A.3 Prompt Templates for Value Pending Evaluation with Show Presence

Value Pending Evaluation with Show Presence Prompt Template: In-Hospital Mortality Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Analyze timestamped observations from an ICU stay and predict whether the patient
will survive their hospital admission.

DATA FORMAT:
Each observation is one of the following:
1.  A triplet:
<time> hours_since_start </time> <channel> channel_name </channel> <value> value </value>

2.  A pair:
<time> hours_since_start </time> <channel> channel_name </channel>

IMPORTANT:
- For some observations, the measurement was made but the value/content is not available
at the time of prediction.  These observations are shown as time-channel pairs.
- When making the prediction, use both the available values/content and the information
conveyed by the timings and channel types themselves | what types of laboratory
measurements were ordered and what notes were recorded, when, in what sequence, and how
frequently | and what these patterns might signal about the patient's clinical trajectory
and mortality risk.

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) SURVIVAL
(B) MORTALITY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (SURVIVAL): <answer> A </answer>
Example 2 (MORTALITY): <answer> B </answer>

[OBSERVATIONS]
...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will experience in-hospital
mortality.
Follow the output format exactly.  Start directly with <answer>.
```

Value Pending with Show Presence Evaluation Prompt Template: Length-of-Stay Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

Prompt Template (continued)

```
TASK: Given timestamped observations from the first 24-hour window of an ICU stay, predict
whether the patient will ultimately have a short ICU stay (<96 hours) and survive.

DATA FORMAT:
Each observation is one of the following:
1.  A triplet:
<time> hours_since_start </time> <channel> channel_name </channel> <value> value </value>

2.  A pair:
<time> hours_since_start </time> <channel> channel_name </channel>

CLASS DEFINITION:
- SHORT_STAY: ICU stay < 96 hours and survived.
- LONG_STAY: ICU stay >= 96 hours or death.

IMPORTANT:
- For some observations, the measurement was made but the value/content is not available
at the time of prediction.  These observations are shown as time-channel pairs.
- When making the prediction, use both the available values/content and the information
conveyed by the timings and channel types themselves - what types of laboratory
measurements were ordered and what notes were recorded, when, in what sequence, and
how frequently - and what these patterns might signal about whether the patient's ICU
course is likely to stabilize quickly (short stay and survive) versus become prolonged or
deteriorate (ICU stay >= 96 hours or death).

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) LONG_STAY
(B) SHORT_STAY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (LONG_STAY): <answer> A </answer>
Example 2 (SHORT_STAY): <answer> B </answer>

[OBSERVATIONS]
...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will have a short ICU stay
and survive.
Follow the output format exactly.  Start directly with <answer>.
```

A.4 Prompt Templates for Channel-Ordered Serialization

In channel-ordered serialization, observations are grouped into channel sections that follow a fixed channel order, and remain time-ordered within each section. The channel identity is carried by the section header rather than repeated in every observation, so each record reduces to a time-value pair.

Channel-Ordered Serialization Prompt Template: In-Hospital Mortality Prediction

System

Prompt Template (continued)

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Analyze timestamped observations from an ICU stay and predict whether the patient
will survive their hospital admission.

DATA FORMAT: Observations are grouped by channel in predefined order.  Each channel
section begins with [Channel:  channel_name] and contains the time-ordered observations
for that channel, where each observation is:
  <time> hours_since_start </time> <value> value </value>

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) SURVIVAL
(B) MORTALITY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (SURVIVAL): <answer> A </answer>
Example 2 (MORTALITY): <answer> B </answer>

[OBSERVATIONS]
[Channel:  Anion Gap]
<time> hours_since_start </time> <value> value </value>
...

[Channel:  Bicarbonate]
<time> hours_since_start </time> <value> value </value>
...

...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will experience in-hospital
mortality.
Follow the output format exactly.  Start directly with <answer>.
```

Channel-Ordered Serialization Prompt Template: Length-of-Stay Prediction

System

```
You are a critical care physician predicting outcomes from ICU monitoring data including
laboratory measurements and radiology notes.
```

User

```
TASK: Given timestamped observations from the first 24-hour window of an ICU stay, predict
whether the patient will ultimately have a short ICU stay (<96 hours) and survive.

DATA FORMAT: Observations are grouped by channel in predefined order.  Each channel
section begins with [Channel:  channel_name] and contains the time-ordered observations
for that channel, where each observation is:
```

Prompt Template (continued)

```
   <time> hours_since_start </time> <value> value </value>

CLASS DEFINITION:
- SHORT_STAY: ICU stay < 96 hours and survived.
- LONG_STAY: ICU stay >= 96 hours or death.

OUTPUT FORMAT (FOLLOW EXACTLY):
Provide your answer in <answer> </answer> tags.
Write ONLY a single letter:  A or B

(A) LONG_STAY
(B) SHORT_STAY

Do NOT include any explanatory text outside the tags.  Start directly with <answer>.

Example 1 (LONG_STAY): <answer> A </answer>
Example 2 (SHORT_STAY): <answer> B </answer>

[OBSERVATIONS]
[Channel:  Anion Gap]
<time> hours_since_start </time> <value> value </value>
...

[Channel:  Bicarbonate]
<time> hours_since_start </time> <value> value </value>
...

...
[/OBSERVATIONS]

Based on the observations above, predict whether this patient will have a short ICU stay
and survive.
Follow the output format exactly.  Start directly with <answer>.
```

B Dataset Details

B.1 EHR Database Overview

MIMIC-IV. The Medical Information Mart for Intensive Care IV (MIMIC-IV) (Johnson et al., 2023c; 2024; 2023b) is a publicly available EHR dataset sourced from the Beth Israel Deaconess Medical Center (BIDMC) in Boston, MA, covering hospital and ICU admissions from 2008 to 2022. MIMIC-IV is organized into three modules. The `hosp` module contains hospital-wide data, including laboratory measurements, microbiology cultures, medication orders, billing, and patient tracking. The `icu` module contains ICU bedside data, including charted observations, infusions, and patient outputs, recorded with both a clinically relevant timestamp (`charttime`) and a database availability timestamp (`storetime`). The `note` module contains deidentified free-text clinical notes, including discharge summaries and radiology reports authored by physicians. Data were deidentified in accordance with the HIPAA Safe Harbor provision and are available via PhysioNet.

eICU. The eICU Collaborative Research Database (eICU-CRD) v2.0 (Pollard et al., 2018; 2019) is a publicly available multi-center critical care database built from data collected through the Philips eICU Program, a telehealth ICU monitoring system in which remote care teams monitor patients at participating hospitals via the eCareManager platform. The database spans admissions from 2014 to 2015 across hundreds of ICU units at hospitals throughout the United States. Available data include vital signs, laboratory measurements, medications, APACHE severity scores, admission diagnoses, and care plan documentation. The care plan is documented using structured multiple-

choice lists covering care provider types, prognosis, treatment status, goals of care, and end-of-life planning. Unlike the rich physician-authored free-text notes in MIMIC-IV, eICU care plan entries are comparatively templated and less linguistically varied. The database is deidentified per HIPAA Safe Harbor and available via PhysioNet.

B.2 Dataset Extraction

We describe the extraction pipeline for each database separately, covering cohort definition, channel selection, filtering criteria, and label construction. Both databases use a 24-hour observation window from ICU admission and a 70%/15%/15% random train/validation/test split at the ICU stay level.

B.2.1 MIMIC-IV

Numerical Channels. We extract 21 laboratory tests from the `labevents` table in the MIMIC-IV `hosp` module: Glucose, Sodium, Chloride, Creatinine, Urea Nitrogen, Bicarbonate, Anion Gap, Hemoglobin, Hematocrit, Magnesium, Platelet Count, Phosphate, White Blood Cells, Calcium (Total), MCH, Red Blood Cells, MCHC, MCV, RDW, Neutrophils, and Vancomycin. Each lab measurement is timestamped by its `charttime`, which records the clinically relevant time of specimen collection. The `storetime`, which records when the result became available in the database, is also retained for the value pending evaluation in Section 5.5.

Text Channel. We extract radiology notes from MIMIC-IV-Note (Johnson et al., 2023b), which contains deidentified free-text radiology reports authored by physicians. Each note carries a `charttime` and a `storetime`. Discharge summaries are excluded as they may contain explicit outcome information. Radiology reports do not have this concern.

Cohort and Filtering. For both tasks, we restrict observations to the first 24 hours of the ICU stay. We require each stay to have at least 64 lab measurements and between 2 and 6 notes within this window, and retain only stays satisfying both modality requirements. The lower bound on note count ensures that notes form a temporal series rather than an isolated observation, while the upper bound removes ICU stays with an anomalously high number of notes to prevent context length explosion.

IHM Task. The cohort consists of ICU stays with a length of stay (LOS) of at least 24 hours. The label is in-hospital mortality, derived from the `hospital_expire_flag` field in the admissions table.

LOS Task. The cohort consists of ICU stays with an LOS of at least 48 hours. Following Han et al. (2024), the binary label is 1 for short stay, defined as an ICU stay shorter than 96 hours with the patient surviving, and 0 otherwise.

B.2.2 eICU

Numerical Channels. We extract 24 laboratory tests from the lab table: glucose, potassium, sodium, chloride, creatinine, BUN, bicarbonate, anion gap, Hgb, Hct, magnesium, platelets, phosphate, WBC, calcium, MCH, RBC, MCHC, MCV, RDW, neutrophils, and Vancomycin (random, trough, and peak). Timestamps are derived from `labresultoffset`, which records minutes elapsed since admission and is converted to hours.

Text Channel. We extract care plan notes from the `note` table, which stores clinical documentation entered through the eCareManager telehealth platform. Each row in the table represents a single form field selection, with `notepath` encoding a full hierarchy of section, field label, and field value (e.g., `notes/Progress Notes/Interventions/Major/Sepsis - evaluation and management`), and `notevalue` storing the authoritative leaf value. We define a note event as all rows sharing the same patient stay, timestamp (`noteoffset`), and note type (e.g., `Brief Progress`, `Admission`, `Intubation`), and exclude CPR-type notes, as their presence may constitute label leakage for the mortality prediction task.

Because the eCareManager system interleaves clinical documentation with UI control fields (e.g., print settings, view mode toggles, sign-off confirmations), many rows in the table are structural artifacts rather than clinical observations. We filter out these noise rows, specifically those whose

Table 5: Dataset statistics.

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
# MITS Samples	3,404	2,896	3,667	2,712
Positive Rate (%)	25.79	27.83	22.42	37.13
# Labs Per MITS	$81.9_{\pm 17.7}$	$82.2_{\pm 17.8}$	$36.4_{\pm 18.8}$	$36.8_{\pm 19.0}$
# Notes Per MITS	$3.12_{\pm 1.17}$	$3.15_{\pm 1.17}$	$3.86_{\pm 1.23}$	$3.87_{\pm 1.24}$

section corresponds to system UI paths (e.g., `View and Save`), whose field label corresponds to scaffolding controls (e.g., `View Options`, `Save Options`, `Sign As`), or whose field value is a generic modal selection (e.g., `System View`, `Performed`, `Not Significant`).

For the remaining informative rows, we format each field differently depending on its section. In the `Interventions` and `Assessment and Plan` sections, the field value itself is the full clinical problem or intervention name (e.g., `Respiratory failure - evaluation and management`), so we emit only the value. In all other sections, the field label names a clinical attribute and the value records the finding, so we emit a `field_label: field_value` token (e.g., `Smoking Status: denies smoking`). All informative tokens from a note event are then deduplicated and concatenated into a single note string prefixed with its note type. For example, a `Brief Progress` note event might produce:

```
[Brief Progress]
Communication with other healthcare providers and/or family
Respiratory failure - evaluation and management
```

and an `Admission` note event might produce:

```
[Admission]
Smoking Status: denies smoking
Ethanol Use: rare
```

Note events whose reconstructed text is empty after noise removal are discarded.

Cohort and Filtering. We restrict observations to the first 24 hours of the ICU stay. We require each stay to have at least 16 lab measurements and between 3 and 10 care plan note events within this window, and retain only stays satisfying both modality requirements.

IHM Task. The cohort consists of ICU stays with a LOS of at least 24 hours. The binary label is in-hospital mortality, defined as `hospitaldischargestatus == "Expired"`.

LOS Task. The cohort consists of ICU stays with a LOS of at least 48 hours. The binary label is short stay, defined as an ICU stay shorter than 96 hours with the patient surviving at unit discharge (`unitdischargestatus == "Alive"`).

B.3 Dataset Statistics and Analyses

We present the basic statistics of the 4 datasets in Table 5, including the number of MITS samples, positive rate, number of lab measurements per MITS, and number of clinical notes per MITS. We further report the number of lab measurements per sample for each channel in Table 6. We also report the character count per note for each dataset to highlight differences in note properties between MIMIC-IV and eICU. The note length distributions are shown in Fig. 4. Note lengths in both MIMIC-IV and eICU exhibit a long-tailed distribution. Notes in MIMIC-IV-IHM and MIMIC-IV-LOS can reach up to more than $5{,}000$ characters, whereas those in eICU-IHM and eICU-LOS reach only about $1{,}000$ characters, reflecting the richer free-text content of MIMIC-IV compared to eICU.

C Model and Hyper-Parameter Configurations

We present the model and hyper-parameter configurations of MILM and the baselines below. All hyper-parameters are selected by validation AU-ROC.

Table 6: Lab measurement counts per sample per channel. For eICU, Vancomycin (*) aggregates the random, trough, and peak sub-types.

	MIMIC-IV		eICU	
Channel	**IHM**	**LOS**	**IHM**	**LOS**
Anion Gap	4.18±1.39	4.17±1.39	1.73±1.37	1.75±1.39
Bicarbonate	4.21±1.38	4.20±1.37	1.99±1.28	2.02±1.30
Blood Urea Nitrogen (BUN)	4.22±1.35	4.21±1.35	1.98±1.27	2.00±1.27
Calcium, Total	3.78±1.74	3.76±1.73	1.93±1.27	1.95±1.28
Chloride	4.33±1.36	4.32±1.36	2.01±1.29	2.04±1.30
Creatinine	4.22±1.36	4.21±1.35	1.98±1.26	2.00±1.26
Glucose	3.43±4.35	3.57±4.46	2.03±1.36	2.06±1.39
Hematocrit (Hct)	5.28±1.70	5.31±1.74	1.90±1.30	1.93±1.30
Hemoglobin (Hgb)	1.91±2.59	2.00±2.68	1.94±1.35	1.95±1.32
Magnesium	3.98±1.55	3.97±1.54	1.29±1.17	1.33±1.19
MCH	4.73±1.49	4.75±1.51	1.64±1.01	1.66±1.00
MCHC	4.74±1.49	4.75±1.51	1.65±1.01	1.66±1.00
MCV	4.74±1.49	4.75±1.50	1.65±1.01	1.66±1.01
Neutrophils (-polys)	0.77±0.89	0.79±0.90	0.69±0.83	0.70±0.85
Phosphate	3.80±1.72	3.78±1.72	0.97±1.13	1.00±1.14
Platelet Count	4.96±1.70	4.98±1.73	1.67±1.03	1.68±1.02
Potassium	—	—	2.26±1.49	2.28±1.49
Red Blood Cells (RBC)	4.74±1.49	4.75±1.50	1.65±1.00	1.67±1.00
RDW	4.73±1.49	4.75±1.51	1.50±1.04	1.53±1.05
Sodium	4.27±1.47	4.27±1.47	2.17±1.52	2.19±1.51
Vancomycin*	0.13±0.38	0.13±0.38	0.07±0.28	0.07±0.28
White Blood Cells (WBC)	4.74±1.49	4.76±1.51	1.65±1.00	1.67±0.99

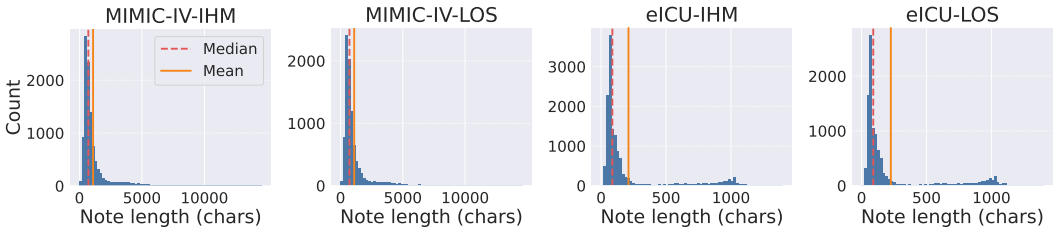


Figure 4: Distribution of note character lengths across all four datasets. Dashed red and solid orange vertical lines indicate the median and mean, respectively.

C.1 MILM

We fine-tune with QLoRA (Dettmers et al., 2023) (rank $r = 16$, $\alpha = 16$, dropout 0.05) on the query, key, value, and output projection matrices. We use AdamW (Loshchilov & Hutter, 2019) with a learning rate of 10^{-4}, cosine schedule, warmup ratio 0.05, and effective batch size 8. Stage 1 (value-redaction fine-tuning) trains for 8 epochs across all datasets. The number of Stage 2 epochs is dataset-dependent, as shown below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
# Stage 2 Epochs	5	10	8	10

C.2 Baselines

For each baseline, we search over the space in Table 7 and select the configuration with the best validation AU-ROC. The number of training epochs is selected by validation AU-ROC as well. Hyper-parameters outside the searched set follow the values in the original implementations.

Table 7: Hyper-parameter search space for each baseline. Selected values are reported in the corresponding paragraphs and tables below.

Method	Search Space
GRU-D	`hidden_size` $\in \{32, 64, 128\}$, `mlp_hidden` $\in \{32, 64\}$, `dropout` $\in \{0.1, 0.3\}$
mTAND	`learning_rate` $\in \{5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$, `embed_dim` $\in \{32, 64, 128\}$
KEDGN	`hidden_dim` $\in \{4, 8, 12\}$, `learning_rate` $\in \{1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}\}$
UTDE	`learning_rate` $\in \{5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$, `embed_dim` $\in \{32, 64, 128\}$, `layers` $\in \{2, 3\}$, `cross_layers` $\in \{2, 3\}$
FuseMoE	`learning_rate` $\in \{1 \times 10^{-4}, 2 \times 10^{-4}\}$, `embed_dim` $\in \{32, 64\}$
VITAL-stats	`d_ff` $\in \{32, 64, 128\}$
VITAL-LLM	`n_prototypes` $\in \{256, 512, 1000\}$, `learning_rate` $\in \{1 \times 10^{-3}, 2 \times 10^{-3}, 4 \times 10^{-3}\}$, `dropout` $\in \{0.1, 0.2\}$, `d_ff` $\in \{32, 64, 128, 256\}$
ISTS-PLM	`learning_rate` $\in \{5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}\}$

GRU-D. GRU-D (Che et al., 2018) extends the standard Gated Recurrent Unit with two trainable temporal decay mechanisms for irregularly sampled time series with missing values. Specifically, an input decay γ_x decays each unobserved feature toward its empirical mean as a function of the elapsed time since its last observation, while a hidden state decay γ_h similarly decays the GRU hidden state. The binary missingness mask is additionally fed into the GRU gates as an informative signal.

We use `hidden_size` $= 32$ followed by an MLP with `mlp_hidden` $= 64$. `dropout` and `recurrent_dropout` are both set to 0.1. The model is trained with the Adam optimizer (Kingma & Ba, 2015) at a learning rate of 10^{-3}, with L_2 weight decay of 10^{-4} on the MLP hidden layer weights and gradient clipping at norm 1.0. The batch size is 32. The number of training epochs is dataset-dependent, as shown below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
`epochs`	200	200	150	150

mTAND. mTAND (Shukla & Marlin, 2021) models irregularly sampled time series by learning a continuous-time embedding of observation timestamps and using a multi-head attention mechanism to interpolate observed values onto a fixed set of reference time points, producing a fixed-length representation without requiring imputation or temporal discretization. For classification, we use the encoder-only variant (mTAND-Enc), where the mTAND module maps irregular observations to K reference-point embeddings that are then aggregated by a GRU, followed by a two-layer residual projection head.

We use `embed_dim` $= 64$, a time embedding dimension of 32, 8 attention heads, $K = 24$ reference points (matching the 24-hour observation window), and a GRU hidden size of 128. Dropout is set to 0.3. The model is trained with the Adam optimizer with gradient clipping at norm 1.0 and a batch size of 32 for 250 epochs across all datasets. The learning rate is dataset-dependent, as shown below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
`learning_rate`	2×10^{-4}	1×10^{-4}	2×10^{-4}	1×10^{-4}

KEDGN. KEDGN (Luo et al., 2024) is a graph neural network for irregularly sampled time series, conditioned on the textual knowledge of each variable. A frozen BERT (Devlin et al., 2019) encodes each variable's text description into a semantic representation, which is projected into a q-dimensional query vector that indexes a shared weight pool, giving each variable its own parameter space, and into an n-dimensional node embedding whose pairwise cosine similarities form a complete variable correlation graph. At each timestamp, the graph is masked to the observed variables and its edge weights are rescaled by a density-aware term that compares the sampling density of the two endpoints, scaled by α. Both modules are injected into a graph convolutional recurrent network that updates a variable's hidden state only at its observed timestamps, avoiding imputation.

We use $q = 7$, $n = 7$, $\alpha = 2$, Adam optimizer, batch size 32, and gradient clipping at 1.0 for 50 epochs. Dataset-specific hyper-parameters are summarized below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
`hidden_dim`	8	12	8	8
`learning_rate`	5×10^{-3}	5×10^{-3}	1×10^{-3}	1×10^{-3}

UTDE. UTDE (Zhang et al., 2023b) is a multimodal framework for irregular EHR time series that unifies two complementary embedding strategies via a learned gate: an imputation-based embedding (forward-fill followed by a 1-D convolution) and an mTAND-based interpolation embedding, yielding a unified time series representation. Clinical notes are cast as multivariate irregular time series and encoded through a separate mTAND module. The two modalities are then fused through J identical layers of interleaved self-attention and cross-attention.

We use the Adam optimizer with `learning_rate` $= 2 \times 10^{-4}$, batch size 32, gradient clipping at 1.0, dropout 0.3, time embedding dimension 32, 8 attention heads, `layers` $= 2$ self-attention layers, and `cross_layers` $= 2$ cross-attention layers. Dataset-specific hyper-parameters are summarized below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
`embed_dim`	64	64	32	64
`num_of_notes`	5	5	9	9
`epochs`	150	150	250	250

FuseMoE. FuseMoE (Han et al., 2024) shares the per-modality encoding backbone with UTDE, using mTAND for irregularly sampled time series and a projection layer for clinical note embeddings, but replaces the cross-attention fusion with a sparse Mixture-of-Experts (MoE) fusion layer. In each layer, a joint router receives the concatenated embeddings of all modalities and dispatches each sample to the top-K of E expert MLPs.

We use Adam with `learning_rate` $= 2 \times 10^{-4}$, batch size 32, gradient clipping at 1.0, dropout 0.3, `embed_dim` $= 64$, time embedding dimension 32, 8 attention heads, 2 self-attention layers, 2 cross-encoder layers, 3 experts, top-K=2 routing, a joint router, MoE hidden size 512, and a load-balance auxiliary loss with coefficient 0.01. Dataset-specific hyper-parameters are summarized below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
`num_of_notes`	5	5	9	9
`epochs`	100	100	150	150

VITAL-stats & VITAL-LLM. VITAL (Kwon et al., 2025) is a variable-aware framework that differentiates between two types of clinical variables: frequently measured vital signs processed through LLM reprogramming, and infrequently measured lab tests embedded via per-channel summary statistics or a learnable `[Not measured]` token. Since the numerical channels of our datasets contain only laboratory measurements, we evaluate both branches separately: VITAL-stats applies the lab branch, representing each channel by a four-dimensional summary vector (mean, median, min, max) followed by feature mixing. VITAL-LLM applies the vital sign reprogramming branch, which requires regular-grid input and is therefore applied to an hourly-discretized version of the lab channels. For fairness of comparison, VITAL-LLM uses the same backbone LLM as MILM (Qwen3-4B-Instruct-2507).

Both variants are trained with Adam, `learning_rate` $= 1 \times 10^{-3}$, `dropout` $= 0.1$, and gradient clipping at 1.0. VITAL-stats uses batch size 128 and `d_ff` $= 32$ uniformly. VITAL-LLM uses batch size 8 with 16 gradient accumulation steps (effective batch size 128), `n_prototypes` $= 1000$, 6 reprogramming attention heads, and bfloat16 precision. Dataset-specific hyper-parameters are summarized in Tables 8 and 9.

Table 8: Dataset-specific hyper-parameters for VITAL-stats.

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
epochs	50	100	50	100

Table 9: Dataset-specific hyper-parameters for VITAL-LLM.

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
d_ff	32	64	32	64

ISTS-PLM. ISTS-PLM (Zhang et al., 2025) adopts a series-based representation of irregularly sampled time series, processing each variable's observations as a univariate sequence through two successive pretrained language model (PLM) stages. The first stage is a time-aware PLM that replaces the absolute positional embeddings with learnable continuous-time embeddings, enabling the model to reason about intra-series temporal dynamics. The second stage is a variable-aware PLM that replaces positional embeddings with learnable variable embeddings, capturing inter-series correlations across asynchronous channels. We use GPT-2 (Radford et al., 2019) as the time-aware PLM and BERT (Devlin et al., 2019) as the variable-aware PLM, following the original paper. Applying Qwen3 is not feasible because Qwen3 uses rotary position embeddings (RoPE) (Su et al., 2024), which are fused into the attention computation and cannot be straightforwardly replaced by continuous-time embeddings. We use the first 6 layers of each PLM ($d_{\text{model}} = 768$), follow the original paper to freeze all PLM parameters except layer normalization, use dropout 0.1, Adam optimizer, batch size 6, and gradient clipping at 1.0 for 40 epochs. Dataset-specific hyper-parameters are summarized below:

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
learning_rate	2×10^{-3}	1×10^{-3}	1×10^{-3}	5×10^{-4}

Timestamp-to-Text Fusion (TTF) Module. The Timestamp-to-Text Fusion (TTF) module, introduced in Time-IMM (Chang et al., 2025), addresses the challenge of integrating asynchronously timestamped clinical notes with numerical time series by producing a temporally-aware and fixed-size text representation. We adopt the T2V-XAttn variant. Each note embedding is concatenated with a Time2Vec encoding (Kazemi et al., 2019) of its timestamp, projected to a common dimension, and then attended to by a single learned query vector via single-head cross-attention, yielding a pooled text representation that is conditioned on both the semantic content and the temporal position of each note. This representation is concatenated to the representation of numerical observations before the classification head. We apply TTF to augment GRU-D, mTAND, KEDGN, VITAL-stats, VITAL-LLM, and ISTS-PLM, producing the corresponding "+ Note" baseline variants reported in the main table. All TTF encoders use 768-dimensional BioBERT (Lee et al., 2020) note embeddings. We use the 6 most recent notes per patient on MIMIC-IV and the 10 most recent notes per patient on eICU.

D Compute Resources

All baseline experiments are run on an NVIDIA GeForce RTX 3090 (24 GB). MILM requires more GPU memory due to LLM fine-tuning, and is therefore run on an NVIDIA H100 GPU (96 GB) on a GH200 node. This overhead reflects the per-layer LoRA fine-tuning that genuinely adapts the LLM's weights to domain-specific temporal knowledge and gives MILM the capacity to predict under either setting: from sampling patterns alone or from both patterns and values. In contrast, the LLM-based irregular time series baselines keep the LLM weights frozen by design and train surrounding modules. We report total wall-clock training time per run in Tables 10 and 11, and peak GPU memory usage in Table 12.

Table 10: Total wall-clock training time (minutes) for baselines, run on an NVIDIA RTX 3090.

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
GRU-D	15.00	12.88	8.02	4.43
+ Note	17.12	14.70	7.22	5.43
mTAND	3.82	3.22	2.98	2.23
+ Note	6.05	5.33	5.90	4.25
KEDGN	3.37	3.02	1.43	1.35
+ Note	4.50	3.45	3.00	2.37
UTDE	6.20	5.55	11.87	11.98
FuseMoE	5.03	5.13	11.86	11.82
VITAL-stats	0.50	0.85	0.35	0.50
+ Note	0.78	1.40	0.72	1.00
VITAL-LLM	428.78	372.51	352.32	121.15
+ Note	698.17	359.30	465.97	167.53
ISTS-PLM	8.67	23.06	8.42	9.62
+ Note	25.70	21.63	18.97	18.60

Table 11: Total wall-clock training time (minutes) for MILM, run on an NVIDIA H100.

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
MILM (Stage 1)	254.38	207.07	306.73	206.18
MILM (Stage 2)	162.00	271.57	311.45	212.18

Table 12: Peak GPU memory usage (GB).

	MIMIC-IV-IHM	MIMIC-IV-LOS	eICU-IHM	eICU-LOS
GRU-D	0.03	0.03	0.02	0.02
+ Note	0.11	0.11	0.10	0.10
mTAND	0.38	0.58	0.20	0.12
+ Note	0.34	0.63	0.31	0.23
KEDGN	0.30	0.33	0.29	0.29
+ Note	0.36	0.37	0.36	0.36
UTDE	0.59	0.62	0.95	0.79
FuseMoE	0.85	0.85	1.17	1.19
VITAL-stats	0.03	0.03	0.03	0.03
+ Note	0.12	0.12	0.12	0.12
VITAL-LLM	11.16	11.18	11.22	11.24
+ Note	11.27	11.29	11.35	11.37
ISTS-PLM	4.87	2.86	3.13	2.81
+ Note	4.89	3.75	3.31	2.46
MILM (Stage 1)	48.96	44.99	26.48	34.68
MILM (Stage 2)	64.20	71.20	39.93	52.43

E Value Pending Evaluation Details

E.1 Dataset Statistics and Analyses

For the value pending evaluation in Section 5.5, an observation's value is considered pending if its `storetime`, the timestamp at which the result was available in the database, exceeds the 24-hour prediction time. When `storetime` is absent, we conservatively fall back to `charttime`, treating the observation as available on time. The same rule is applied to radiology notes.

Table 13 reports the average number and rate of pending observations per ICU stay. On average, each stay has approximately 2.74 (IHM) and 2.71 (LOS) pending lab measurements, representing about 3.3% of all lab observations. Notes are pending at a higher rate: 13.5% (IHM) and 14.0% (LOS). Table 14 shows the average and median `storetime` − `charttime` delay per channel. Most lab delays cluster around 0.6 to 1.3 hours depending on the panel, while note delays are markedly longer (median $\approx$ 3.2 hours), consistent with the higher note pending rate.

Table 13: Average number (left) and rate (right) of pending observations per ICU stay.

	MIMIC-IV-IHM	MIMIC-IV-LOS
Lab	2.74	2.71
Note	0.43	0.45

	MIMIC-IV-IHM	MIMIC-IV-LOS
Lab	3.33%	3.30%
Note	13.53%	14.02%

Table 14: Average and median `storetime` − `charttime` delay (hours) per channel across all ICU stays.

	MIMIC-IV-IHM		MIMIC-IV-LOS	
Channel	Avg.	Median	Avg.	Median
Anion Gap	1.26	1.13	1.25	1.12
Bicarbonate	1.24	1.12	1.23	1.10
Calcium, Total	1.20	1.08	1.20	1.08
Chloride	1.22	1.10	1.22	1.08
Creatinine	1.22	1.08	1.21	1.08
Glucose	0.09	0.05	0.09	0.05
Hematocrit	0.61	0.52	0.61	0.52
Hemoglobin	0.10	0.07	0.10	0.07
MCH	0.62	0.53	0.61	0.52
MCHC	0.62	0.53	0.61	0.52
MCV	0.62	0.53	0.61	0.52
Magnesium	1.21	1.08	1.20	1.08
Neutrophils	1.27	0.87	1.26	0.87
Phosphate	1.21	1.10	1.21	1.08
Platelet Count	0.66	0.55	0.66	0.53
RDW	0.62	0.53	0.61	0.52
Red Blood Cells	0.62	0.53	0.61	0.52
Sodium	1.23	1.10	1.22	1.10
Urea Nitrogen	1.19	1.08	1.19	1.08
Vancomycin	2.31	1.58	2.31	1.59
White Blood Cells	0.64	0.53	0.64	0.53
Radiology Note	5.73	3.17	5.74	3.23

Figure 5 shows how these counts and rates vary across individual ICU stays. Both are right-skewed: the majority of stays have few or no pending observations, but a nontrivial tail of stays can have 10 to 30% of their observations pending.

Figure 6 shows $p(\texttt{charttime} \mid \text{value pending})$ and $p(\texttt{charttime} \mid \text{value not pending})$ separately for labs and notes. The pending distributions are visibly right-shifted relative to the non-pending distributions, confirming that value-pending observations typically fall near the end of the 24-hour observation window, where recent measurements tend to be most informative for outcome prediction.

E.2 Full Results

We present the full value pending evaluation results in Table 15, extending Table 4 to include baselines with the drop observation countermeasure. Note that it is not straightforward to make the baselines show the presence of value-pending observations without showing the values, as these models require a numeric value for every observed entry and cannot process `NaN` inputs. Qwen3-4B and MILM are able to adopt the show presence countermeasure due to the flexibility of text-based representations.

Value pending evaluation has a larger impact on natively multimodal methods such as UTDE, Qwen3-4B, and MILM since the note channel has a higher pending rate than lab channels (Table 13). However, the two-stage training of MILM effectively mitigates this challenge. By training the model to predict solely from sampling patterns, Stage 1 develops two complementary capabilities. First, by learning to extract predictive signals without value tokens, MILM makes better use of the remaining observations under drop observation when some note content is missing. Second, by learning to treat time-channel patterns as signals in their own right, MILM makes better use of the additional sampling information conveyed under show presence. Both capabilities translate directly to stronger performance than MILM-Direct under value pending evaluation. MILM achieves the best AU-ROC

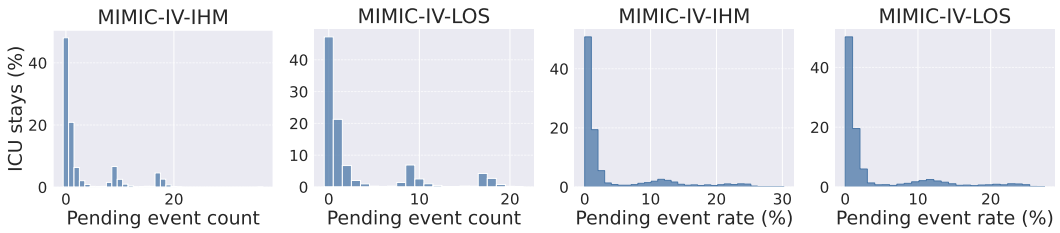


Figure 5: Distribution of pending event count (left two) and pending event rate (right two) per ICU stay at the 24-hour prediction time.

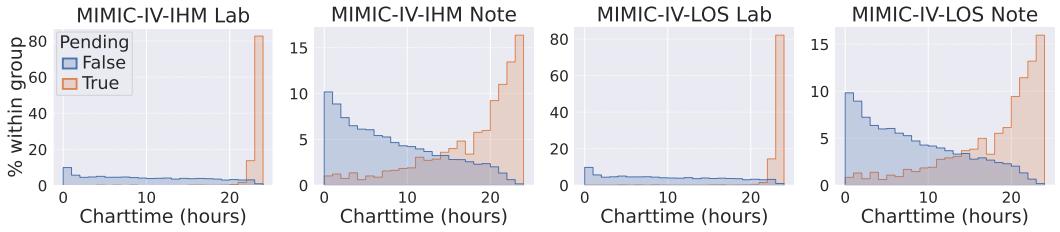


Figure 6: Distribution of `charttime` for value-pending and non-pending observations.

Table 15: Full value pending evaluation results. Best results are **boldfaced** and second-best results are underlined.

		MIMIC-IV-IHM		MIMIC-IV-LOS	
Countermeasure	**Method**	AU-ROC	AP	AU-ROC	AP
Drop Observation	GRU-D	75.07±1.03	52.61±1.07	69.14±1.20	46.37±1.43
	+ Note	73.60±1.79	50.21±4.12	70.12±2.56	47.51±2.46
	mTAND	76.56±1.14	54.63±1.70	71.19±0.83	49.83±1.41
	+ Note	77.46±1.55	56.54±1.05	73.02±1.02	52.86±2.83
	KEDGN	78.42±1.68	58.17±3.53	71.91±1.38	48.66±2.16
	+ Note	71.59±3.88	46.05±5.21	62.36±5.76	39.03±4.56
	UTDE	74.22±2.93	52.54±3.02	73.89±3.17	52.80±3.16
	FuseMoE	75.49±1.47	55.00±0.92	74.56±0.46	55.10±1.75
	VITAL-stats	76.44±0.84	53.14±2.13	73.70±0.95	52.58±2.72
	+ Note	77.10±0.63	52.86±1.99	74.24±1.25	54.47±4.03
	VITAL-LLM	67.96±1.61	40.66±2.00	68.77±1.59	47.70±3.36
	+ Note	67.98±2.77	40.42±3.07	68.80±1.26	48.37±1.83
	ISTS-PLM	79.95±0.59	59.31±0.77	73.93±0.73	54.27±1.59
	+ Note	79.52±0.53	58.22±0.64	73.63±0.51	52.46±0.92
	Qwen3-4B	68.33±0.00	44.06±0.00	67.39±0.00	44.37±0.00
	MILM-Direct	79.04±0.91	59.23±1.63	77.36±1.42	57.71±2.13
	MILM	79.49±0.76	60.02±1.57	**78.65±0.41**	**58.02±0.68**
Show Presence	Qwen3-4B	67.40±0.00	39.80±0.00	66.54±0.00	43.50±0.00
	MILM-Direct	79.21±0.65	58.14±1.37	76.47±1.40	56.19±2.36
	MILM	**80.34±0.64**	**61.23±1.62**	78.14±0.47	57.11±0.96

and AP on both datasets, despite operating under a larger train-test discrepancy than the unimodal baselines.

F MedGemma vs. Qwen

Due to the domain of interest, it is natural for us to consider using large language models pretrained for the biomedical and healthcare domains, such as MedGemma (Sellergren et al., 2025). MedGemma is a collection of medical vision-language foundation models built on the Gemma 3 architecture, designed to interpret and reason about medical images and text across modalities including radiology,

dermatology, histopathology, and ophthalmology. In our preliminary experiments, we compare the zero-shot performance of the 4B instruction-tuned variants of both MedGemma and Qwen3. We report the results below:

	MIMIC-IV-IHM			MIMIC-IV-LOS		
Model	Unparseable (%)	AU-ROC	AP	Unparseable (%)	AU-ROC	AP
MedGemma-4B	62.70	57.81	36.27	89.66	52.66	33.23
Qwen3-4B	**0.00**	**69.82**	**48.36**	**0.00**	**67.98**	**44.82**

	eICU-IHM			eICU-LOS		
Model	Unparseable (%)	AU-ROC	AP	Unparseable (%)	AU-ROC	AP
MedGemma-4B	65.70	52.47	22.40	80.64	53.38	43.33
Qwen3-4B	**0.00**	**64.53**	**29.43**	**0.00**	**55.71**	**45.58**

A large portion of the responses generated by MedGemma-4B are unparseable. For example, instead of generating `<answer> A </answer>`, it may generate `A`, violating the output format specified in the prompt. In contrast, the responses from Qwen3-4B are all parseable, demonstrating its stronger instruction-following capability and stability. We evaluate AU-ROC and AP by assigning a neutral score of 0.5 to unparseable responses to maintain a consistent evaluation cohort. Despite being pretrained on biomedical and healthcare domain data, MedGemma-4B underperforms Qwen3-4B on all 8 metric-dataset combinations. We suspect this is because MedGemma's domain-specific training is primarily oriented toward medical image understanding, with text supervision derived from relatively small medical QA datasets, whereas Qwen3 is pretrained on a substantially larger and more diverse corpus of 36 trillion tokens across 119 languages, which may better support the text representation inputs used in our task. Given the stronger instruction-following capability and better overall performance of Qwen3-4B, we select it as the base LLM for this work.

G Experiments on a Non-Healthcare Dataset

Table 16: Results on the Yelp restaurant closure prediction dataset. Best results are **boldfaced** and second-best results are underlined.

	Yelp	
Method	AU-ROC	AP
mTAND	50.57±1.55	22.71±1.32
+ Note	55.58±2.41	25.47±1.90
KEDGN	53.35±2.44	24.07±1.60
+ Note	54.58±0.99	25.47±0.29
UTDE	<u>64.35±0.98</u>	<u>33.62±0.63</u>
FuseMoE	64.09±1.88	32.79±2.35
ISTS-PLM	51.40±1.78	25.22±1.24
+ Note	50.17±4.00	23.95±1.98
Qwen3-4B	51.58±0.00	25.58±0.00
MILM	**68.18±1.31**	**41.08±2.09**

We conduct an auxiliary experiment on a non-healthcare dataset to evaluate MILM's generality. We build a restaurant closure prediction dataset from the Yelp Open Dataset[1], which releases business records together with the reviews and tips that users posted about them. Each sample is one restaurant, and the observation window is the calendar year 2017. The numerical channel is the star rating attached to each review. The text channel is the free-text body of the reviews and tips. The label is whether the restaurant is closed as of the dataset snapshot, roughly four years after the window closes. Informative sampling has a natural analogue in this setting. Reviews and tips act as a proxy for customer traffic, so how often they arrive and how that rate trends over the year carries signal about whether the restaurant survives, independent of the ratings and the review content.

[1] https://business.yelp.com/data/resources/open-dataset/

We keep businesses whose categories include `Restaurants` and that received at least one review and one tip within the window. We further require between 6 and 50 total observations, and require activity after the window ends so that every restaurant was still operating when the window closed. We subsample the resulting cohort to $5{,}000$ restaurants. Splits follow the $70\%/15\%/15\%$ protocol used for the clinical datasets, grouped by normalized business name so that locations of the same chain do not straddle splits. Aggregate business fields that postdate the observation window, such as the lifetime average rating, the total review count, and the current opening hours, are excluded to prevent leakage. Since the domain is no longer biomedical, we replace BioBERT with BERT (Devlin et al., 2019) as the note embedding model for the baselines. We evaluate a representative subset of the baselines, selecting the strongest performers from each of the three classes described in Section 5.1: mTAND and KEDGN, the two best non-LLM-based methods by average rank in Table 1, UTDE and FuseMoE among the multimodal methods, and ISTS-PLM, which leads VITAL-LLM by a wide margin among the LLM-based methods, together with zero-shot Qwen3-4B.

Table 16 reports the results. MILM performs best on both metrics, with a margin of 3.83 AU-ROC and 7.46 AP over the strongest baseline. The two natively multimodal baselines follow, with UTDE at 64.35 AU-ROC and FuseMoE at 64.09, both well ahead of the rest of the field. Every remaining method performs close to random. The gap is wider than on the clinical datasets because this dataset has a single numerical channel, which places more weight on how a method handles the text channel. These results indicate that MILM is not tied to the healthcare setting that motivated it. Wherever observations arrive irregularly across numerical and textual channels, and the act of observing carries signal, the same serialization and two-stage fine-tuning recipe applies without modification.

H Asset Licenses

Datasets. MIMIC-IV (Johnson et al., 2023c; 2024), MIMIC-IV-Note (Johnson et al., 2023b), and the eICU Collaborative Research Database (Pollard et al., 2018; 2019) are available via PhysioNet under the PhysioNet Credentialed Health Data License 1.5.0, and are used in compliance with their respective data use agreements. The Yelp Open Dataset used in Appendix G is provided by Yelp Inc. under the Yelp Dataset Terms of Use (last updated July 7, 2023), which grants a royalty-free, non-exclusive license for non-commercial academic use by educational institutions and prohibits redistribution of the data or any derivative thereof. We use it solely for non-commercial academic research and release our preprocessing and modeling code but no portion of the dataset.

Models. Qwen3-4B-Instruct-2507 (Yang et al., 2025), BioBERT (Lee et al., 2020), and BERT (Devlin et al., 2019) are released under the Apache License 2.0. GPT-2 (Radford et al., 2019) is released under the Modified MIT License. MedGemma (Sellergren et al., 2025) is released under the Health AI Developer Foundations Terms of Use. All pretrained models are accessed via the Hugging Face Transformers library (Wolf et al., 2020), which is also released under the Apache License 2.0.