CLINICAL IRREGULAR TIME-SERIES CLASSIFICATION VIA OPERA: OBSERVATION-PROCESS ENCODING AND RESIDUAL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In clinical irregular multivariate time series (IMTS), how often measurements are taken, when they occur, and which variables are recorded provide predictive information beyond measured values. However, existing approaches often couple observation modeling with specific architectures, limiting the flexibility to combine monitoring information with the distinct modeling strengths of different IMTS backbones. We propose Observation-Process Encoding and Residual Adaptation (OPERA), a lightweight plug-in module for clinical IMTS classification. Its *Observation-Process Encoder* uses observation times and variable identities, but not measured values, to construct human-readable descriptors of monitoring amount, timing, and variable selection, making monitoring patterns explicit and analyzable. The *Residual Observation Gate* then adaptively adds an observation-based correction to the backbone's patient representation, guided by both the monitoring pattern and the backbone representation. This patient-level interface preserves the original encoding and aggregation mechanisms, allowing the same design to complement heterogeneous IMTS backbones. Experiments on public intensive care unit (ICU) benchmarks show that OPERA consistently improves all five evaluated backbones at only marginal additional computational cost and achieves the best aggregate performance among representative methods.

1 INTRODUCTION

Clinical irregular multivariate time series (IMTS), especially intensive care unit (ICU) records, support predictions for individual patients across high-stakes tasks such as mortality prediction, phenotype classification (Harutyunyan et al., 2019), sepsis prediction, and physiological deterioration detection (Reyna et al., 2020; Hyland et al., 2020). Unlike regularly sampled signals, ICU records are sparse, asynchronous, and selectively observed, with only a subset of variables measured at non-uniform time points, which makes clinical IMTS classification challenging.

Most existing IMTS models address this challenge by focusing on representing measured physiological values under irregular sampling—a *value-centric* perspective. One line of work converts irregular observations into aligned sequences through discretization, imputation, or interpolation (Shukla & Marlin, 2019), enabling standard sequence encoders to operate on a fixed grid. Another line models irregular observations more directly, including recurrent models with masks, elapsed times, or decay mechanisms (Baytas et al., 2017; Che et al., 2018), continuous-time models that evolve hidden states between observations (Rubanova et al., 2019; Kidger et al., 2020), attention-based architectures (Shukla & Marlin, 2021), graph-based architectures (Zhang et al., 2022), patch-based architectures (Luo et al., 2025), and hybrid graph-attention designs (Wang et al., 2026). Despite their architectural differences, these methods, as aforementioned, prioritize measured-value modeling under irregular and incomplete sampling.

However, irregular sampling is not just an obstacle to modeling measured values. In ICU data, observation patterns reflect both patient state and care processes: some patients are monitored more frequently, selected variables are repeatedly measured, and tests concentrate at particular stages of care (Sisk et al., 2021). Consequently, *how often measurements are taken*, *when they occur*, and *which variables are recorded* provide predictive information beyond measured values. This

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.

relates to "informative missingness", where the presence or absence of measurements is statistically associated with patient state.

Existing methods incorporate these cues in two main ways. Many IMTS models use masks, time gaps, timestamps, or observed entries within measured-value encoders rather than separately summarizing each patient's overall monitoring pattern. Other methods directly encode sampling events or records with measured values removed using dedicated event or language-model encoders (Karami et al., 2024; Chung et al., 2026). In both cases, observation modeling is often coupled with particular architectures, limiting the flexibility to combine monitoring information with the distinct modeling strengths of different backbones, such as modeling continuous-time dynamics in ODE-RNN (Rubanova et al., 2019) or cross-variable dependencies in Raindrop (Zhang et al., 2022). This motivates a reusable way to summarize monitoring patterns and combine them with different backbones while retaining their original encoding and aggregation mechanisms.

In response, we propose Observation-Process Encoding and Residual Adaptation (OPERA), a lightweight plug-in module for clinical IMTS classification. As shown in Figure 1, OPERA uses two parallel branches: an existing IMTS backbone that encodes the measured clinical series and an Observation-Process Encoder that summarizes monitoring patterns. Using observation times and variable identities, but not measured values, this encoder constructs human-readable descriptors of observation intensity and coverage, temporal allocation and burstiness, and variable selection and diversity. These descriptors provide an explicit, analyzable summary of each patient's monitoring pattern. The backbone produces a patient representation—a single feature vector summarizing the input record for classification. The Residual Observation Gate adds an observation-based correction to this vector, using both the encoded monitoring summary and the backbone representation to account for the clinical context of monitoring. This patient-level interface preserves the original encoding and aggregation mechanisms across heterogeneous backbones. Experiments on public ICU benchmarks demonstrate consistent improvements across all five evaluated backbones with marginal additional computational cost and the best aggregate performance among representative methods.

Our contributions are summarized as follows:

- OPERA provides a lightweight plug-in framework that complements measured-value modeling with explicit monitoring-pattern encoding. Its residual gate adds monitoring information to the patient representation produced by each backbone. The same design supports different backbones without redesigning their encoders or aggregation mechanisms.
- The Observation-Process Encoder uses observation times and variable identities, but not measured values, to construct human-readable descriptors of monitoring amount, timing, and variable selection. These descriptors make each patient's monitoring pattern explicit and analyzable, providing a complementary signal for clinical prediction.
- Experiments on three public ICU benchmarks demonstrate the standalone predictive value of observation descriptors, consistent improvements across five heterogeneous backbones with marginal additional computational cost, and the best aggregate performance among representative methods under a unified evaluation protocol.

2 Related Work

Irregular Multivariate Time-Series Modeling. Clinical IMTS modeling has been studied through a broad range of representation strategies. Early methods transform irregular observations into aligned sequences through discretization, imputation, or interpolation (Shukla & Marlin, 2019). Other approaches directly model irregular observations, including recurrent architectures with elapsed-time or decay mechanisms (Baytas et al., 2017; Che et al., 2018) and continuous-time models that evolve hidden states between observations (Rubanova et al., 2019; Kidger et al., 2020). More recent methods introduce set-based aggregation (Horn et al., 2020), multi-time attention (Shukla & Marlin, 2021), self-supervised Transformers over observation triplets (Tipirneni & Reddy, 2022), graph modeling (Zhang et al., 2022), temporal warping (Zhang et al., 2023), and patch-based or hybrid architectures (Luo et al., 2025; Wang et al., 2026). Despite their architectural differences, these methods mainly use observation-related inputs to improve representations of the measured clinical series. OPERA instead summarizes each patient's monitoring pattern

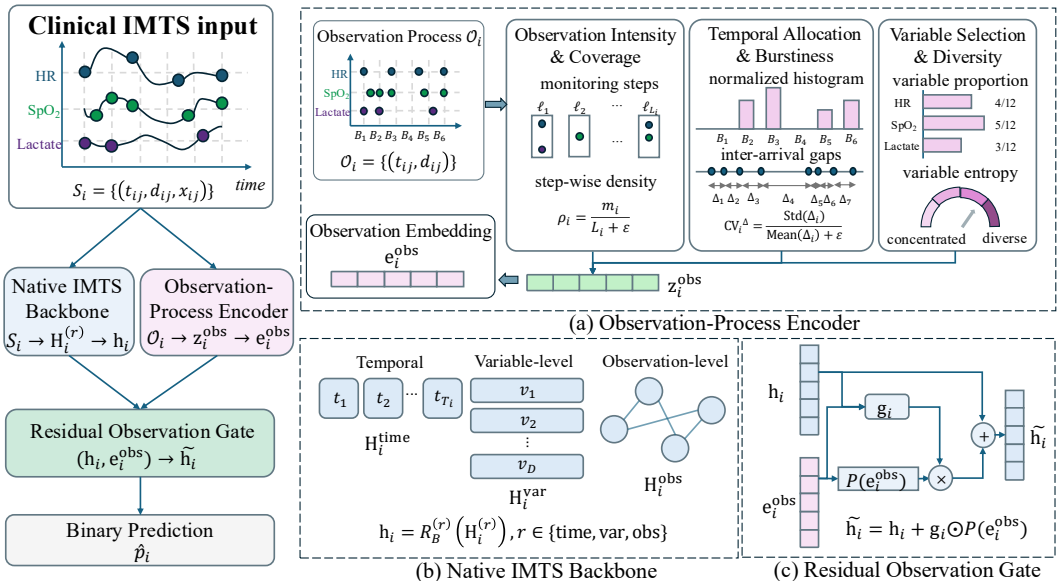

Figure 1: Framework overview of OPERA. The native IMTS branch uses the backbone's original encoder and readout to produce a patient representation. The Observation-Process Encoder summarizes observation times and variable identities without measured values. The Residual Observation Gate uses both representations to add an observation-based correction before binary prediction. Panels (a)–(c) detail the observation encoder, native backbone, and residual gate, respectively.

from observation times and variable identities, providing an explicit representation that complements measured-value modeling.

Informative Missingness and Observation Process. Clinical missingness and measurement patterns are often informative rather than purely random (Rubin, 1976; Cismondi et al., 2013). Accordingly, many IMTS models incorporate masks, elapsed times, timestamps, or observed entries into recurrent, continuous-time, attention-based, and graph-based value encoders (Che et al., 2018; Rubanova et al., 2019; Shukla & Marlin, 2021; Zhang et al., 2022). Although effective, these cues usually remain auxiliary inputs within the same architecture that models measured values, rather than forming an explicit representation of the overall monitoring pattern. More direct approaches separately encode sampling events or records with measured values removed. For example, TEE4EHR and MILM learn dedicated representations from these observation records (Karami et al., 2024; Chung et al., 2026). Such approaches often couple observation modeling with specific architectures, limiting the flexibility to combine monitoring information with the distinct modeling strengths of different IMTS backbones. OPERA instead summarizes each patient's monitoring pattern and integrates the resulting representation through a reusable gated residual interface.

Backbone-Compatible Adaptation. Recent work has also explored reusable components for heterogeneous irregular time-series models. MissTSM develops an imputation-free, missing-feature-aware layer that can be combined with different time-series architectures (Neog et al., 2026), while ReIMTS characterizes the temporal, variable-level, and observation-level representations exposed by heterogeneous IMTS models (Li et al., 2026). OPERA complements this direction by explicitly summarizing monitoring patterns without using measured values. Its gated residual interface operates on backbone patient representations, preserving native encoders and aggregation mechanisms while supporting different internal representations.

3 METHODOLOGY

This section develops OPERA around two requirements: explicitly summarizing each patient's monitoring pattern and integrating the resulting representation across different IMTS backbones.

We first formulate the task and then present the **Observation-Process Encoder** and **Observation-Process Adaptation**, implemented through the Residual Observation Gate.

3.1 Problem Definition

For the i-th patient, let $\mathcal{S}_i = \{(t_{ij}, d_{ij}, x_{ij})\}_{j=1}^{m_i}$ denote the time-ordered collection of observed events, where t_{ij} is the observation time, $d_{ij} \in \{1, \ldots, D\}$ is the variable index, x_{ij} is the measured value, and m_i is the number of observed variable–time entries. We consider binary patient-level classification, where the goal is to estimate the probability of the positive class:

$$\hat{p}_i = f(\mathcal{S}_i), \tag{1}$$

where $\hat{p}_i$ denotes the predicted probability of the positive class. Retaining observation times and variable identities, but not measured values, gives the observation process $\mathcal{O}_i = \{(t_{ij}, d_{ij})\}_{j=1}^{m_i}$. Our goal is to encode $\mathcal{O}_i$ and use the resulting monitoring representation to complement the patient representation learned by an existing IMTS backbone.

3.2 Framework Overview

The design of OPERA follows a simple principle: separate how a patient is monitored from what values are measured, and combine the two through their patient representations. As shown in Figure 1, this yields two parallel encoding branches followed by gated residual adaptation.

The native branch retains the original IMTS backbone and its native readout, preserving its established mechanism for representing physiological dynamics. In parallel, the Observation-Process Encoder uses observation times and variable identities, but not measured values, to summarize monitoring amount, timing, and variable selection. This explicit side pathway makes each patient's monitoring pattern analyzable and provides complementary information for prediction.

Once both branches have produced patient representations, the Residual Observation Gate adds an observation-based correction to the backbone representation, guided by both representations. This common interface avoids aligning the observation branch with individual time steps, variables, or observations, while preserving each backbone's internal structure and modeling strengths.

In essence, OPERA concentrates observation-specific modeling in a compact auxiliary pathway while retaining the backbone's original encoding and aggregation mechanisms. This design provides both an explicit monitoring representation and a reusable interface for heterogeneous IMTS backbones. The following sections detail observation-process encoding and residual adaptation.

3.3 Observation-Process Encoder

As illustrated in Figure 1(a), the Observation-Process Encoder maps the value-free observation process to a patient-level embedding in two stages. First, a deterministic extractor summarizes monitoring patterns using human-readable descriptors. A lightweight embedding network then maps these descriptors to a compact representation for residual adaptation.

The descriptors are organized into three groups: *observation intensity and coverage*, *temporal allocation* and burstiness, and *variable selection and diversity*. These groups characterize how much, when, and which variables are observed, respectively. We introduce representative descriptors below; the complete descriptor set and boundary conventions are provided in Appendix A.

We first group observations in $\mathcal{O}_i$ by timestamp. Each distinct timestamp with at least one observed variable defines a monitoring step. For patient i, this gives $\{(\bar{\tau}_{i\ell}, \mathcal{V}_{i\ell})\}_{\ell=1}^{L_i}$, ordered by time, where L_i is the number of monitoring steps, $\bar{\tau}_{i\ell} \in [0, 1]$ is the normalized timestamp, and $\mathcal{V}_{i\ell}$ is the set of variables observed at that step. We write $r_{i\ell} = |\mathcal{V}_{i\ell}|$ for the number of variables observed at step ℓ. Throughout this section, $\epsilon > 0$ is a small constant for numerical stability.

Observation Intensity and Coverage The same total number of observations can be spread over many monitoring steps or concentrated in a few steps. We therefore describe monitoring amount

using the total number of observed entries and the average number of variables observed per step:

$$m_i = \sum_{\ell=1}^{L_i} r_{i\ell}, \qquad \rho_i = \frac{m_i}{L_i + \epsilon}. \tag{2}$$

Here, m_i counts observed variable–time entries, while ρ_i measures the average measurement density at a monitoring step. Together with L_i, these quantities distinguish the number of monitoring steps from the number of variables observed at each step.

Temporal Allocation and Burstiness Monitoring volume and density do not specify when observations occur within the input window. We divide the normalized window $[0, 1]$ into B equal-width bins $\{\mathcal{B}_b\}_{b=1}^{B}$. The proportion of monitoring steps in bin b is

$$p_{ib} = \frac{\sum_{\ell=1}^{L_i} \mathbb{I}(\bar{\tau}_{i\ell} \in \mathcal{B}_b)}{L_i + \epsilon}. \tag{3}$$

The resulting profile $\mathbf{p}_i = [p_{i1}, \ldots, p_{iB}]$ summarizes the distribution of monitoring steps across the window. Each step contributes once, regardless of how many variables are observed at that step.

To capture temporal irregularity within this coarse allocation, we consider the gaps between consecutive monitoring steps. Let $\Delta_{i\ell} = \bar{\tau}_{i,\ell+1} - \bar{\tau}_{i\ell}$ for $\ell = 1, \ldots, L_i - 1$ denote the inter-arrival gaps. Their coefficient of variation is

$$\mathrm{CV}_i^{\Delta} = \frac{\mathrm{Std}\Big(\{\Delta_{i\ell}\}_{\ell=1}^{L_i-1}\Big)}{\mathrm{Mean}\Big(\{\Delta_{i\ell}\}_{\ell=1}^{L_i-1}\Big) + \epsilon}. \tag{4}$$

A larger coefficient indicates more uneven spacing between monitoring steps, whereas a smaller coefficient indicates more regular spacing. Together, the occupancy profile and gap variability characterize temporal allocation and burstiness.

Variable Selection and Diversity Patients with similar temporal monitoring patterns may still differ in which variables are measured. To capture this allocation of measurement effort, let $n_{i,d}$ be the number of observations of variable d and define $q_{i,d} = n_{i,d}/(m_i + \epsilon)$. The vector $\mathbf{q}_i = [q_{i,1}, \ldots, q_{i,D}]$ records the share of all observed entries contributed by each variable.

We summarize the diversity of this allocation using normalized variable entropy:

$$\mathrm{Ent}_i^{\mathrm{var}} = -\frac{1}{\log D} \sum_{d=1}^{D} q_{i,d} \log(q_{i,d} + \epsilon). \tag{5}$$

Higher entropy indicates a more even allocation of observations across variables, whereas lower entropy indicates concentration on fewer variables. Thus, $\mathbf{q}_i$ retains variable-specific allocation patterns, while $\mathrm{Ent}_i^{\mathrm{var}}$ summarizes their overall diversity.

Finally, we concatenate the descriptors into $\mathbf{z}_i^{\mathrm{obs}}$ and map them to a compact embedding $\mathbf{e}_i^{\mathrm{obs}} = \phi_\xi(\mathbf{z}_i^{\mathrm{obs}})$ for residual adaptation. Before entering the embedding network, the descriptor coordinates are standardized using training-split statistics and clipped, as specified in Appendix A. The raw descriptors can be computed once per record and reused across backbones, while the embedding network is learned for each backbone–task configuration.

3.4 Observation-Process Adaptation

Having encoded the observation process, we next integrate its embedding with the backbone's native patient representation. OPERA connects these two streams through gated residual adaptation after the native readout. This placement allows observation information to complement the backbone representation while retaining the original encoding and aggregation mechanisms, providing a common adaptation interface across heterogeneous IMTS backbones.

Backbone Representation Interface For neural backbones, the adaptation interface must accommodate different forms of internal representation. As illustrated in Figure 1(b), these may be organized by time step, clinical variable, or individual observation. Let $\mathbf{H}_i^{(r)}$ denote the final structured representation, where $r \in \{\text{time}, \text{var}, \text{obs}\}$. The backbone's native readout produces a patient-level vector:

$$\mathbf{h}_i = R_B^{(r)}\left(\mathbf{H}_i^{(r)}\right). \tag{6}$$

Although their internal structures differ, these backbones all expose a patient-level representation after readout. We therefore apply observation adaptation at this common interface, avoiding the need to align observation descriptors with backbone-specific temporal, variable, or observation axes. The original encoding and readout mechanisms are retained.

Residual Observation Gate To combine the observation embedding with the native patient representation, we first match their dimensionalities using a learnable projection $\overline{\mathbf{e}}_i^{\text{obs}} = P(\mathbf{e}_i^{\text{obs}}) \in \mathbb{R}^{d_h}$, where d_h is the dimensionality of $\mathbf{h}_i$. The projected embedding provides an observation-based correction that can be added to the native representation.

The relevance of a monitoring pattern may depend on the clinical context captured by the backbone. We therefore condition a dimension-wise gate on both the native representation and the observation embedding:

$$\mathbf{g}_i = \sigma\big(W_g[\mathbf{h}_i; \mathbf{e}_i^{\text{obs}}] + \mathbf{b}_g\big) \in (0,1)^{d_h}, \tag{7}$$

where W_g and $\mathbf{b}_g$ are learnable parameters and $\sigma(\cdot)$ is the element-wise sigmoid function. Each gate entry scales the corresponding component of the projected observation embedding, allowing the correction to depend jointly on the monitoring pattern and the native patient representation.

The gated correction is then added to the native representation:

$$\widetilde{\mathbf{h}}_i = \mathbf{h}_i + \mathbf{g}_i \odot \overline{\mathbf{e}}_i^{\text{obs}}, \tag{8}$$

where $\odot$ denotes element-wise multiplication. The additive structure retains a direct path from the native representation to the classifier, while the observation branch learns a complementary correction. Together, this common interface and gated residual update incorporate monitoring information without redesigning the backbone's encoding or aggregation mechanisms.

3.5 Prediction and Training Objective

The classifier maps the adapted patient representation to binary class probabilities, $\boldsymbol{\pi}_i = \text{softmax}(C_\omega(\widetilde{\mathbf{h}}_i))$, where C_ω outputs two logits and $\hat{p}_i = \pi_{i,1}$ is the predicted positive-class probability. Given a batch of N patients with labels $y_i \in \{0,1\}$, we minimize the cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N} \log \pi_{i,y_i}. \tag{9}$$

The native backbone, observation embedding network, projection, gate, and classifier are trained jointly under this objective, while descriptor extraction remains deterministic. Joint training allows the observation correction and the native representation to adapt together to the prediction task.

4 Experimental Study

4.1 Experimental Setup

Datasets. We evaluate OPERA on three public ICU benchmarks. For **P12**, derived from the PhysioNet 2012 Challenge dataset (Silva et al., 2012), we use clinical observations from the first 48 hours for in-hospital mortality prediction. For **P19**, derived from the PhysioNet 2019 Challenge dataset (Reyna et al., 2020), we use the first 48 hours of observations to predict the first sepsis onset during the subsequent 48 hours. For **MIMIC-III** (Johnson et al., 2016), we use an admission-level in-hospital mortality dataset constructed with the GARLIC preprocessing pipeline (Wang et al., 2026). Detailed cohort construction and preprocessing procedures are provided in Appendix B.

Table 1: Predictive value of observation descriptors and plug-in gains of OPERA. Results are test AUROC/AUPRC (%; mean ± standard deviation over three seeds). HGB-Value uses measured-value summaries, HGB-Obs uses observation descriptors, and HGB-Joint combines both. +OPERA adapts the preceding backbone. Avg. Δ reports gains averaged across datasets (percentage points), relative to HGB-Value or the native backbone. Bold marks the best mean within each block.

Method	**P12**		**P19**		**MIMIC-III**		**Avg. Δ**	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
HGB-Value	87.83±0.00	58.82±0.00	70.17±0.00	14.92±0.00	89.64±0.10	65.61±0.24	—	—
HGB-Obs	76.79±0.00	36.19±0.00	72.33±0.00	16.06±0.00	85.33±0.04	54.55±0.13	—	—
HGB-Joint	**87.96**±0.00	**59.96**±0.00	**72.60**±0.00	**18.98**±0.00	**90.13**±0.07	**66.99**±0.11	+1.02	+2.19
RNN-Decay	86.88±0.18	56.95±0.49	72.12±0.63	17.48±0.37	89.86±0.16	65.98±0.23	—	—
+OPERA	**87.06**±0.08	**58.29**±0.64	**74.87**±0.71	**21.07**±1.55	**90.31**±0.26	**67.07**±0.35	+1.13	+2.01
GRU-D	86.14±0.08	55.90±0.96	67.42±0.97	14.19±1.32	89.52±0.40	64.81±0.82	—	—
+OPERA	**86.99**±0.41	**57.62**±1.10	**73.14**±0.76	**17.73**±0.37	**89.87**±0.27	**66.43**±0.32	+2.31	+2.29
ODE-RNN	84.13±0.33	50.22±0.62	66.05±0.67	11.37±1.43	89.21±0.18	64.03±0.27	—	—
+OPERA	**84.78**±0.90	**52.02**±1.43	**73.28**±0.17	**18.74**±1.19	**89.37**±0.19	**65.28**±0.25	+2.68	+3.47
Hi-Patch	85.90±0.90	55.87±0.59	70.67±2.17	14.64±1.08	89.80±0.39	66.21±0.82	—	—
+OPERA	**86.41**±0.41	**57.37**±1.08	**72.04**±1.11	**15.21**±0.69	**90.46**±0.31	**67.53**±1.09	+0.85	+1.13
Raindrop	83.88±0.45	49.80±0.65	66.29±1.06	10.77±0.64	87.42±0.35	59.35±1.50	—	—
+OPERA	**84.41**±0.43	**51.62**±0.78	**72.77**±0.53	**14.23**±0.39	**88.74**±0.28	**63.04**±0.73	+2.78	+2.99

Backbones and Comparison Methods. We organize the evaluation into three groups: descriptor diagnostics, backbone plug-in tests, and comparisons with representative methods. **(1) HGB diagnostics.** We use histogram-based gradient boosting (HGB) (Friedman, 2001; Pedregosa et al., 2011) to assess the standalone and complementary predictive value of observation descriptors. **(2) Plug-in backbones.** We evaluate Native–OPERA pairs using five heterogeneous neural backbones: RNN-Decay; GRU-D (Che et al., 2018); ODE-RNN (Rubanova et al., 2019); Hi-Patch (Luo et al., 2025); and Raindrop (Zhang et al., 2022). These backbones cover temporal, variable-level, and observation-level representations. **(3) Comparison methods.** For broader comparison, we include recurrent and attention-based models RNN-Δt, RNN-Mean, DA-RNN (Qin et al., 2017), RETAIN (Choi et al., 2016), and mTAN (Shukla & Marlin, 2021); continuous-time models L-ODE-RNN and L-ODE-ODE (Rubanova et al., 2019); graph- and Transformer-based methods MTGNN (Wu et al., 2020), MTSFormer (Zheng et al., 2024), Warpformer (Zhang et al., 2023), and GARLIC (Wang et al., 2026); and missingness- and observation-oriented methods STraTS (Tipirneni & Reddy, 2022), TEE4EHR (Karami et al., 2024), and MissTSM (Neog et al., 2026).

Implementation Details. All experiments are conducted on NVIDIA RTX 4080 SUPER GPUs. Neural models are implemented in PyTorch, while HGB uses scikit-learn. For each dataset, all methods share a fixed stratified 80%/10%/10% training/validation/test split, and all preprocessing and normalization statistics are fitted on the training split only. OPERA uses a $(D+22)$-dimensional descriptor vector with six temporal bins, followed by a two-layer observation embedding network with hidden width 128. Neural models use model-specific optimizers and training schedules. Checkpoints are selected by validation AUPRC, and test AUROC and AUPRC are reported as mean ± standard deviation over three seeds. Detailed implementations are provided in Appendix C.

4.2 Main Results

Table 1 evaluates the predictive value of observation descriptors using HGB and the plug-in effectiveness of OPERA across five heterogeneous neural backbones. Table 2 compares RNN-Decay equipped with OPERA against representative clinical IMTS methods. Three observations emerge. (1) **Observation patterns carry standalone and complementary predictive information.** In Table 1, HGB-Obs predicts outcomes using only observation descriptors, while HGB-Joint outperforms HGB-Value in both metrics on all three datasets. These results show that observation descriptors provide predictive information that complements measured-value summaries. (2) **OPERA**

Table 2: Comparison with representative clinical IMTS methods under a unified protocol. Results are test AUROC/AUPRC (%; mean ± standard deviation over three seeds). Bold and underline indicate the best and second-best results in each column. Ours is RNN-Decay + OPERA.

Method	P12		P19		MIMIC-III	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
RNN-Δt	85.02±0.28	52.42±1.79	63.99±0.43	10.87±0.44	89.40±0.10	65.00±0.21
RNN-Mean	83.99±0.35	55.08±0.82	59.32±1.56	9.07±0.73	87.91±0.33	60.52±0.66
L-ODE-RNN	81.21±0.36	43.54±0.75	65.11±0.57	15.37±2.44	88.87±0.20	64.80±0.29
L-ODE-ODE	83.93±0.42	52.26±0.88	60.89±0.77	10.11±2.03	88.77±0.33	64.01±0.34
mTAN	84.99±1.39	53.27±2.87	62.95±0.90	13.82±2.30	90.22±0.16	66.80±0.42
MTGNN	81.63±1.20	45.35±3.51	59.27±1.35	9.37±1.35	86.87±0.86	55.79±3.14
MTSFormer	85.07±0.27	56.91±1.32	67.11±0.64	12.95±0.36	88.54±0.08	59.40±1.40
Warpformer	<u>86.98±1.61</u>	56.86±2.38	<u>71.40±2.01</u>	<u>16.16±2.07</u>	<u>90.38±0.31</u>	66.99±1.12
DA-RNN	85.77±0.28	54.20±1.89	68.75±1.26	14.70±0.57	89.53±0.14	65.90±0.33
RETAIN	85.59±0.11	<u>57.45±0.46</u>	68.09±0.25	15.01±0.27	89.39±0.36	65.51±0.94
GARLIC	86.12±0.25	56.50±2.42	69.57±2.50	13.93±3.12	88.83±0.24	63.35±1.31
TEE4EHR	86.29±0.38	56.12±1.12	66.12±1.95	13.56±0.94	90.02±0.70	66.46±0.36
MissTSM	83.58±0.52	48.24±0.04	65.48±1.28	12.16±0.18	89.44±0.24	63.64±0.81
STraTS	86.27±0.29	55.83±1.94	68.75±2.84	14.35±2.60	**91.09±0.03**	**69.10±0.06**
Ours	**87.06±0.08**	**58.29±0.64**	**74.87±0.71**	**21.07±1.55**	90.31±0.26	<u>67.07±0.35</u>

provides consistent plug-in gains. It improves both AUROC and AUPRC for every backbone on every dataset, with average gains of 1.95 and 2.38 percentage points, respectively, across the 15 backbone–dataset combinations. The gains span temporal, variable-level, and observation-level representations, demonstrating broad compatibility across heterogeneous architectures. (3) **OPERA achieves the strongest aggregate performance.** As shown in Table 2, RNN-Decay equipped with OPERA ranks first in both metrics on P12 and P19. It also achieves the highest AUROC and AUPRC averaged across the three datasets among the compared methods, while STraTS leads on MIMIC-III. These results show that explicitly exposing complementary monitoring patterns enables a classic recurrent backbone to compete favorably with more specialized native architectures.

4.3 Ablation Studies

Table 3 examines observation-process encoding and residual adaptation using RNN-Decay on three ICU benchmarks. Three findings emerge. (1) **Descriptor groups contribute differently across datasets.** Removing temporal or variable descriptors reduces mean AUPRC on P19 by 2.28 and 3.58 percentage points, respectively, whereas removing intensity descriptors leaves it essentially unchanged. On P12 and MIMIC-III, removing any descriptor group lowers mean AUPRC, supporting the combined use of monitoring amount, temporal allocation, and variable selection. (2) **Structured descriptors outperform the learnable alternative.** Replacing the structured descriptors with a learnable encoder of observation times and variable identities yields lower mean AUPRC on all three datasets, supporting the effectiveness of the explicit descriptor design. (3) **The residual gate improves observation integration.** Directly adding the projected observation embedding to the backbone representation without gating (*w/o Gate*) or using direct concatenation (*Direct Concat*) yields lower

Table 3: Ablation study of observation-process encoding and adaptation with RNN-Decay. Results are test AUPRC (%), reported as mean ± standard deviation over three seeds. The best result for each dataset is shown in bold.

Variant	P12	P19	MIMIC-III
Native	56.95±0.49	17.48±0.37	65.98±0.23
Full OPERA	**58.29±0.64**	21.07±1.55	**67.07±0.35**
w/o Intensity	58.27±0.83	**21.08±0.28**	66.86±0.30
w/o Temporal	57.58±1.09	18.79±1.36	66.87±0.45
w/o Variable	58.14±1.14	17.49±0.08	66.82±0.13
Learnable Encoder	57.23±0.78	17.67±0.10	66.55±1.02
w/o Gate	57.64±0.55	19.26±1.49	66.88±0.71
Direct Concat	57.84±0.47	20.10±3.29	66.97±0.56

mean AUPRC than Full OPERA on all three datasets, with larger gaps on P12 and P19. These comparisons support the benefit of gated residual adaptation over ungated addition and direct concatenation.

4.4 Case Study

Figure 2 presents a matched-pair case study on P12. The mortality-positive and matched negative patients receive nearly identical native GRU-D risks (0.411 vs. 0.415) and have comparable numbers of observed events (548 vs. 546). After incorporating the observation-process signal, their predictions separate in the correct direction: the positive patient's risk increases to 0.789, whereas the negative patient's risk decreases to 0.247. The two patients exhibit markedly different variable-selection profiles. The positive patient has lower normalized variable entropy (0.758 vs. 0.845), indicating that monitoring is more concentrated on a limited subset of variables. In particular, the positive patient shows higher observation proportions for NIBP, GCS, and HR, whereas the negative patient's measurements are more concentrated on urine, temperature, and blood-gas variables. These differences reveal distinct variable-selection patterns despite similar overall observation volumes. This example illustrates how OPERA uses selective monitoring patterns to complement measured-value representations and refine otherwise similar native predictions.

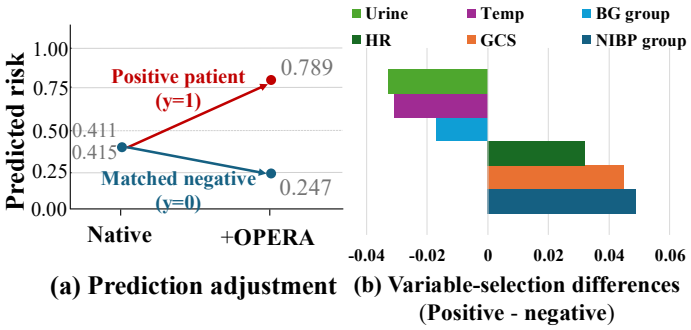

Figure 2: Matched-pair analysis on P12: (a) native and OPERA-adapted GRU-D risks; (b) variable observation proportions (positive minus negative).

4.5 Computational Overhead

To assess the marginal computational overhead of OPERA, we perform paired measurements on three backbones: RNN-Decay, Hi-Patch, and Raindrop. As shown in Table 4, OPERA adds only 69.8K–326.2K trainable parameters and 0.4–6.2 MB of peak allocated memory. Hi-Patch exhibits small runtime increases of 2.1% for training and 1.0% for inference, while the mean runtime changes for RNN-Decay and Raindrop remain below 1%. The limited overhead follows from caching the deterministic descriptors offline and restricting online computation to a compact embedding network and a gated residual correction. These measurements show that OPERA converts observation-process information into consistent predictive gains at only marginal additional computational cost.

Table 4: Computational overhead of OPERA on MIMIC-III. Runtime values are mean relative changes over five alternating paired repeats.

Backbone	**RNN-Decay**	**Hi-Patch**	**Raindrop**
Extra Params	+82.0K	+69.8K	+326.2K
Δ Train	+0.2%	+2.1%	+0.9%
Δ Infer.	+0.5%	+1.0%	+0.1%
Δ Mem.	+1.5 MB	+0.4 MB	+6.2 MB

5 Conclusion

We presented OPERA, a lightweight plug-in module that combines explicit monitoring-pattern encoding with gated residual adaptation. Its Observation-Process Encoder summarizes observation times and variable identities without using measured values. The Residual Observation Gate incorporates this information into backbone patient representations while preserving their original encoding and aggregation mechanisms. Across three public ICU benchmarks, OPERA consistently improves all five evaluated backbones with marginal additional computational cost and achieves the best aggregate performance among representative methods. These results establish explicit monitoring summaries as an effective, reusable complement to heterogeneous clinical IMTS backbones.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript organization, retrieval and discovery of related literature, and research discussions. They were also used to check conference formatting requirements and refine the LaTeX layout, including table and figure placement. The authors take full responsibility for the final manuscript, including the accuracy and integrity of its claims, references, and reported results.

REFERENCES

Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 65–74, 2017. doi: 10.1145/3097983.3097997.

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. doi: 10.1038/s41598-018-24271-9.

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. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3504–3512, 2016.

Hsing-Huan Chung, Shijun Li, Yoav Wald, Xing Han, Suchi Saria, and Joydeep Ghosh. MILM: Large language models for multimodal irregular time series with informative sampling, 2026. URL `https://arxiv.org/abs/2605.13711`.

Federico Cismondi, André S. Fialho, Susana M. Vieira, Shane R. Reti, João M. C. Sousa, and Stan N. Finkelstein. Missing data in medical databases: Impute, delete or classify? *Artificial Intelligence in Medicine*, 58(1):63–72, 2013. doi: 10.1016/j.artmed.2013.01.003.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

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. doi: 10.1038/s41597-019-0103-9.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4353–4363. PMLR, 2020.

Stephanie L. Hyland, Martin Faltys, Matthias Hüser, Xinrui Lyu, Thomas Gumbsch, Cristóbal Esteban, Christian Bock, Max Horn, Michael Moor, Bastian Rieck, Marc Zimmermann, Dean Bodenham, Karsten Borgwardt, Gunnar Rätsch, and Tobias M. Merz. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature Medicine*, 26(3):364–373, 2020. doi: 10.1038/s41591-020-0789-4.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Marzyeh Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035, 2016. doi: 10.1038/sdata.2016.35.

Hojjat Karami, David Atienza, and Anisoara Ionescu. TEE4EHR: Transformer event encoder for better representation learning in electronic health records. *Artificial Intelligence in Medicine*, 154: 102903, 2024. doi: 10.1016/j.artmed.2024.102903.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707, 2020.

Boyuan Li, Zhen Liu, Yicheng Luo, and Qianli Ma. Learning recursive multi-scale representations for irregular multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JEIDxiTWzB.

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-Patch: Hierarchical patch GNN for irregular multivariate time series. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41494–41519. PMLR, 2025.

Abhilash Neog, Arka Daw, Sepideh Fatemi Khorasgani, Medha Sawhney, Aanish Pradhan, Mary E. Lofton, Bennett J. McAfee, Adrienne Breef-Pilz, Heather L. Wander, Dexter W. Howard, Cayelan C. Carey, Paul Hanson, and Anuj Karpatne. Investigating a model-agnostic and imputation-free approach for irregularly-sampled multivariate time-series modeling. *Transactions on Machine Learning Research*, 2026.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2627–2633, 2017. doi: 10.24963/ijcai.2017/366.

Matthew A. Reyna, Chris S. Josef, Russell Jeter, Supreeth P. Shashikumar, M. Brandon Westover, Shamim Nemati, Gari D. Clifford, and Ashish Sharma. Early prediction of sepsis from clinical data: The PhysioNet/computing in cardiology challenge 2019. *Critical Care Medicine*, 48(2): 210–217, 2020. doi: 10.1097/CCM.0000000000004145.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ODEs for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581.

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_.

Ikaro Silva, George Moody, Daniel J. Scott, Leo A. Celi, and Roger G. Mark. Predicting in-hospital mortality of ICU patients: The PhysioNet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, volume 39, pp. 245–248, 2012.

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. doi: 10.1093/jamia/ocaa242.

Sindhu Tipirneni and Chandan K. Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data*, 16(6):105:1–105:17, 2022. doi: 10.1145/3516367.

Ruirui Wang, Yanke Li, Manuel Günther, and Diego Paez-Granados. GARLIC: Graph attention-based relational learning of multivariate time series in intensive care. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4ZAwmIaA9y.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 753–763, 2020. doi: 10.1145/3394486.3403118.

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, 2023. doi: 10.1145/3580305.3599543.

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.

Liangwei Nathan Zheng, Zhengyang Li, Chang George Dong, Wei Emma Zhang, Lin Yue, Miao Xu, Olaf Maennel, and Weitong Chen. Irregularity-informed time series analysis: Adaptive modelling of spatial and temporal dynamics. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3405–3414, 2024. doi: 10.1145/3627673.3679716.

A Complete Observation-Process Descriptor Set

This section completes the descriptor definitions introduced in Section 3.3 and specifies their boundary and preprocessing conventions. We use the same monitoring-step representation $\{(\bar{\tau}_{i\ell}, \mathcal{V}_{i\ell})\}_{\ell=1}^{L_i}$, ordered by time, where $\bar{\tau}_{i\ell} \in [0,1]$ is the normalized timestamp and $\mathcal{V}_{i\ell}$ contains the variables observed at step ℓ. Writing $r_{i\ell} = |\mathcal{V}_{i\ell}|$, the total observation count is $m_i = \sum_{\ell=1}^{L_i} r_{i\ell}$. All descriptors are computed from timestamps and variable identities without accessing measured values.

Observation Intensity and Coverage. We first summarize the amount of monitoring and its extent within the input window:

$$\mathbf{z}_i^{\text{int}} = [m_i,\ L_i,\ \bar{\tau}_{i,L_i},\ \rho_i]\,, \qquad \rho_i = \frac{m_i}{L_i + \epsilon}. \tag{10}$$

These four coordinates record the total number of observed entries, the number of monitoring steps, the last observed time, and the average number of variables observed per step. L_i counts monitoring steps without specifying their temporal spacing.

Temporal Allocation and Burstiness. The preceding quantities do not describe how monitoring is distributed within the window. We therefore retain the temporal occupancy profile $\mathbf{p}_i = [p_{i1}, \ldots, p_{iB}]$ defined in the main paper. Each occupied monitoring step contributes once to this profile, regardless of the number of variables observed at that step.

To additionally capture the temporal distribution of individual observed entries, we weight each monitoring step by $r_{i\ell}$. The resulting observation-time mean and standard deviation are

$$\mu_i^\tau = \frac{\sum_{\ell=1}^{L_i} r_{i\ell}\bar{\tau}_{i\ell}}{m_i + \epsilon}, \qquad \sigma_i^\tau = \left[\frac{\sum_{\ell=1}^{L_i} r_{i\ell}(\bar{\tau}_{i\ell} - \mu_i^\tau)^2}{m_i + \epsilon}\right]^{1/2}. \tag{11}$$

Using the same observation-entry weighting, we summarize the allocation to the first half and the final quarter of the window:

$$a_i^{\text{early}} = \frac{\sum_{\ell=1}^{L_i} r_{i\ell}\mathbb{I}(\bar{\tau}_{i\ell} \le 1/2)}{m_i + \epsilon}, \qquad a_i^{\text{late}} = \frac{\sum_{\ell=1}^{L_i} r_{i\ell}\mathbb{I}(\bar{\tau}_{i\ell} > 3/4)}{m_i + \epsilon}. \tag{12}$$

The half-window contrast compares observation allocation between the first and second halves:

$$c_i^{\text{half}} = \frac{\sum_{\ell=1}^{L_i} r_{i\ell}\left[\mathbb{I}(\bar{\tau}_{i\ell} \le 1/2) - \mathbb{I}(\bar{\tau}_{i\ell} > 1/2)\right]}{m_i + \epsilon}. \tag{13}$$

The sign indicates greater observation allocation to the first (+) or second (−) half of the window.

Finally, we capture local temporal irregularity through inter-arrival gaps. For $L_i > 1$, let $\Delta_{i\ell} = \bar{\tau}_{i,\ell+1} - \bar{\tau}_{i\ell}$ and let $\bar{\Delta}_i$ be their mean. Alongside the inter-arrival coefficient of variation CV_i^Δ defined in the main paper, we use the short-gap fraction:

$$b_i^{\text{short}} = \frac{1}{L_i - 1}\sum_{\ell=1}^{L_i-1} \mathbb{I}\big(\Delta_{i\ell} < \tfrac{1}{2}\bar{\Delta}_i\big)\,. \tag{14}$$

This is the fraction of gaps shorter than half the patient's mean inter-arrival time. The temporal group contains $B + 7$ coordinates: B occupancy proportions, two observation-time moments, two early/late fractions, one half-window contrast, and two gap-based statistics.

Variable Selection and Diversity. Patients with similar temporal monitoring patterns may still differ in which variables are observed. Let $n_{i,d}$ be the observation count of variable d and $q_{i,d} = n_{i,d}/(m_i + \epsilon)$. We retain the variable-selection profile $\mathbf{q}_i = [q_{i,1}, \ldots, q_{i,D}]$ and its normalized variable entropy $\text{Ent}_i^{\text{var}}$, as defined in the main paper.

To complement the allocation profile, variable coverage $\kappa_i = D^{-1}\sum_{d=1}^{D}\mathbb{I}(n_{i,d} > 0)$ records the fraction of variables observed at least once. We further define $s_{i\ell} = r_{i\ell}/D$ as the fraction of variables observed at monitoring step ℓ. The population mean, population standard deviation, and maximum of $\{s_{i\ell}\}_{\ell=1}^{L_i}$ summarize the typical breadth, variability, and peak breadth of simultaneous variable observation. This group contains $D + 5$ coordinates.

Complete Vector. Concatenating the three groups gives the raw descriptor vector $\mathbf{z}_i^{\text{obs}}$ with dimension $F = 4 + (B + 7) + (D + 5) = D + B + 16$. Unless otherwise stated, we use $B = 6$, giving $F = D + 22$: 58 dimensions for P12, 56 for P19, and 125 for MIMIC-III. Table 5 summarizes the complete set.

Boundary and Preprocessing Conventions. Observations sharing a timestamp are grouped into one monitoring step before inter-arrival gaps are constructed. The B temporal bins are left-closed and right-open, except that the final bin includes time 1. Both gap-based coordinates are set to zero when $L_i \leq 1$. All standard deviations use the population definition, and zero-probability entropy terms contribute zero. Padding rows have zero observation masks and are excluded from all descriptor statistics. We use $\epsilon = 10^{-8}$ for numerical stability.

For neural backbones, descriptor coordinates are standardized using the training-split mean and population standard deviation. Zero-variance coordinates use unit scale, and the resulting values are clipped to $[-10, 10]$. Validation and test records use the same training statistics. As in the main paper, this preprocessing is omitted from the notation $\mathbf{e}_i^{\text{obs}} = \phi_\xi(\mathbf{z}_i^{\text{obs}})$. The HGB feature evaluation uses the raw descriptors.

Table 5: Complete observation-process descriptor set. The default is $B = 6$.

Descriptor	Dim.
Observation Intensity and Coverage	
Total observation count	1
Number of monitoring steps	1
Last observed time	1
Average measurement density	1
Temporal Allocation and Burstiness	
Temporal occupancy profile	B
Observation-time mean and standard deviation	2
Early and late allocation fractions	2
Half-window contrast	1
Inter-arrival coefficient of variation	1
Short-gap fraction	1
Variable Selection and Diversity	
Variable-selection profile	D
Normalized variable entropy	1
Variable coverage	1
Active-variable mean, standard deviation, and maximum	3
Total	$D + B + 16$

B Dataset Construction and Preprocessing

Table 6 summarizes the final experimental cohorts. We describe the prediction windows, cohort construction, and input preprocessing for each of the three datasets below.

Table 6: Statistics of the three ICU cohorts. Avg. Obs. is the mean number of observed variable–time entries per sample; Pos.Rate is the positive-class proportion.

Dataset	P12	P19	MIMIC-III
Task	Mortality	Sepsis	Mortality
# Samp.	11,988	9,930	49,377
# Channels	36	34	103
Avg. Obs.	402.3	330.9	395.3
Pos. Rate	14.24%	6.20%	11.64%

P12. P12 is derived from the PhysioNet/Computing in Cardiology Challenge 2012 cohort (Silva et al., 2012). We use clinical observations from the first 48 hours of ICU admission for patient-level in-hospital mortality prediction. The dynamic input contains measurements from 36 clinical variables.

P19. P19 is constructed from the PhysioNet/Computing in Cardiology Challenge 2019 records (Reyna et al., 2020) as a patient-level landmark task rather than the original hourly prediction objective. Because the released positive labels are shifted six hours before clinical onset, onset is recovered as six hours after the first positive label. We retain patients observed at the 48-hour landmark, exclude onset at or before 48 hours, and use only values, masks, and timestamps from the first 48 hours. The target is the first in-stay sepsis onset during $48 < t \leq 96$ hours. Patients discharged before 96 hours without a recorded onset are retained as negative cases up to discharge. Stays without a valid landmark row or dynamic input are excluded. The split is stratified by data source and label, and no post-landmark measurements or timestamps enter the model inputs.

MIMIC-III. We construct an admission-level mortality dataset from MIMIC-III (Johnson et al., 2016) using the GARLIC preprocessing pipeline (Wang et al., 2026). Each sample corresponds to a hospital admission identified by `HADM_ID`. The input window covers the first 48 hours from the first selected event in that admission. The input contains 103 dynamic event-feature channels, including measurement and treatment-related events. Static age and gender fields are stored separately and are not included in the dynamic channel count. The cohort contains 49,377 admissions with 5,747 positive cases; 39,501 admissions are used for training and 4,938 each for validation and testing.

C Experimental Details

C.1 General Experimental Settings

Configuration selection. We prioritize recommended configurations from the original papers or official implementations. When these configurations are unavailable or not directly applicable to our tasks, we select general training hyperparameters, including learning rate, weight decay, and batch size, through grid search on the validation set. The final settings are summarized in Table 7. Training budgets are increased when validation-selected checkpoints reach the epoch limit. The test set is not used to select hyperparameters, random seeds, or checkpoints.

Paired evaluation. Each Native–OPERA pair uses the same frozen cohort, patient-level split, input window, values, masks, timestamps, and normalization statistics. The backbone, hidden dimension, native readout, classifier, optimizer, learning rate, batch size, maximum epochs, early-stopping rule, and random seeds are also matched. OPERA adds only the observation descriptors, embedding network, projection, and residual gate, with no separate tuning of backbone hyperparameters.

Table 7 summarizes the general training hyperparameters. Dataset-dependent values follow the order P12/P19/MIMIC-III; a single value applies to all three datasets. Maximum epochs denotes the training budget, and patience denotes the early-stopping patience in epochs. Model-specific architecture, regularization, and scheduling settings are described below.

Table 7: General training hyperparameters. Triplets follow the order P12/P19/MIMIC-III. PT and FT denote STraTS pretraining and fine-tuning, respectively.

Method	Optimizer	LR	Batch size	Max. epochs	Patience
RNN-Decay	Adam	10^{-3}	256/256/32	40	10
GRU-D	Adam	10^{-3}	256/256/32	40	10
ODE-RNN	Adamax	10^{-3}	64/64/128	80	20
Hi-Patch	Adam	10^{-3}	4/16/4	40	10
Raindrop	Adam	10^{-4}	8	80	10
RNN-Δt	AdamW	10^{-3}	256/256/128	40	10
RNN-Mean	AdamW	10^{-3}	256/256/128	40	10
DA-RNN	AdamW	10^{-3}	256/256/128	40	10
RETAIN	AdamW	10^{-3}	256/256/128	40	10
L-ODE-RNN	Adamax	10^{-3}	64/64/128	80	20
L-ODE-ODE	Adamax	10^{-3}	64/64/128	80	20
mTAN	Adam	10^{-4}	128/256/32	80	20
MTGNN	AdamW	10^{-3}	128/128/64	40	10
MTSFormer	SGD	5×10^{-4}	128/128/64	60	10
Warpformer	Adam	10^{-3}	32/32/16	60	15
GARLIC	Adam	See text	64/256/128	100	10
STraTS (PT)	AdamW	5×10^{-4}	8	30	–
STraTS (FT)	AdamW	5×10^{-5}	8	20	5
TEE4EHR	AdamW	10^{-3}	16	40	10
MissTSM	Adam	10^{-4}	16/16/8	30	10
HGB	–	0.05	–	200 (iter.)	–

C.2 MODEL-SPECIFIC SETTINGS

We summarize each method's modeling approach and core configuration below. Dataset-specific values follow the order P12/P19/MIMIC-III.

OPERA. OPERA summarizes observation times and variable identities and adds a gated residual correction to the backbone's patient representation. The observation embedding uses two fully connected layers of width 128 with ReLU activations. Descriptor construction and preprocessing follow Appendix A; the projection, residual gate, and joint objective follow Sections 3.4 and 3.5.

RNN-Decay. RNN-Decay incorporates a decay mechanism for handling missing observations. The recurrent hidden dimension is 128 on all datasets.

GRU-D. GRU-D (Che et al., 2018) models missingness through decay mechanisms for inputs and hidden states. The recurrent hidden dimension is 128 on all datasets.

ODE-RNN, L-ODE-RNN, and L-ODE-ODE. These models (Rubanova et al., 2019) combine recurrent or latent-state representations with continuous-time dynamics. The configurations use latent dimension 20, recognition dimension 40, `rec_layers`=`gen_layers`=2, and `units`=`gru_units`=64, with the corresponding fields applied to each model's encoder and decoder. The learning rate decays by a factor of 0.999 per optimizer step to a minimum of 10^{-4}.

Hi-Patch. Hi-Patch (Luo et al., 2025) models irregular time series using hierarchical patch graphs. The hidden dimensions are 32/32/16.

Raindrop. Raindrop (Zhang et al., 2022) models irregular multivariate observations using graph-based representations of dependencies between clinical variables.

RNN-Δt. RNN-Δt incorporates observation masks and sampling intervals into its recurrent inputs. The hidden dimension is 128 on all datasets.

RNN-Mean. RNN-Mean uses mean imputation for missing observations before recurrent encoding. The hidden dimension is 128 on all datasets.

DA-RNN. DA-RNN (Qin et al., 2017) uses dual-stage attention to model input-feature and temporal dependencies. The configured hidden dimension is 128 on all datasets.

RETAIN. RETAIN (Choi et al., 2016) uses reverse-time attention to aggregate clinical sequences. The configured hidden dimension is 128 on all datasets.

mTAN. mTAN (Shukla & Marlin, 2021) uses time attention to represent irregularly sampled observations. The recurrent hidden dimension is 256 and the latent dimension is 20 on all datasets.

MTGNN. MTGNN (Wu et al., 2020) combines graph-based and temporal modeling with hidden dimension 128 on all datasets. The P19 implementation uses attention dimension 64.

MTSFormer. MTSFormer (Zheng et al., 2024) models temporal, variable, and irregularity information using a Transformer-based architecture. The hidden dimensions are 300/300/256. SGD uses momentum 0.9 with a constant learning rate.

Warpformer. Warpformer (Zhang et al., 2023) learns multiscale representations through temporal warping and attention. The model dimensions are 64/64/32. On P12 and P19, the inner dimension is 64, key and value dimensions are both 8, and the model uses one attention head and three layers. The learning rate is halved every ten epochs.

GARLIC. GARLIC (Wang et al., 2026) combines relational modeling with classification and reconstruction objectives. The hidden dimensions are 128/512/128. Separate Adam parameter groups use classification learning rates of 10^{-4}, 5×10^{-5}, and 10^{-5} on P12, P19, and MIMIC-III, respectively, and a reconstruction learning rate of 10^{-3}. The plateau scheduler uses maximization mode, reduction factor 0.5, patience 3, and minimum learning rate 10^{-5}.

STraTS. STraTS (Tipirneni & Reddy, 2022) embeds continuous values, timestamps, and variable identities, followed by a Transformer encoder and attention pooling. The hidden dimension is 64, with two layers, 16 attention heads, and dropout and attention dropout both set to 0.2. Pretraining uses next-mark forecasting with MSE; fine-tuning starts from pretrained weights and uses binary cross-entropy with logits and positive-class weight 1.0. The gradient clipping threshold is 0.3.

TEE4EHR. The adapted TEE4EHR implementation (Karami et al., 2024) uses next-mark prediction as an auxiliary task for outcome classification. The auxiliary loss weight is 0.05.

MissTSM. MissTSM (Neog et al., 2026) models time series with missing observations. We use its MAE-based classification configuration and direct fine-tuning for the target outcome.

HGB. HGB uses histogram-based gradient boosting for tabular classification. The *Value*, *Obs*, and *Joint* settings use `HistGradientBoostingClassifier`; observation descriptors enter directly as raw features without a neural embedding network.

D Sensitivity to Temporal Binning

The temporal allocation profile divides the normalized 48-hour input window into B equal-width bins. The default setting $B = 6$ therefore corresponds to eight-hour intervals. To examine sensitivity to this discretization choice, we vary $B \in \{3, 6, 12\}$, corresponding to temporal resolutions of 16, 8, and 4 hours, respectively, on P19 and MIMIC-III. All remaining model, data, and optimization settings are held fixed. The default setting $B = 6$ achieves the highest mean AUPRC on both datasets and the highest mean AUROC on P19. On MIMIC-III, AUROC varies by only 0.16 percentage points across the tested resolutions, while the maximum AUPRC differences are 0.44 and 0.25 points on P19 and MIMIC-III, respectively. Overall, performance remains comparable across the tested temporal resolutions.

Table 8: Sensitivity to the number of temporal bins B with GRU-D. Results are test AUROC/AUPRC (%), reported as mean ± standard deviation over three seeds. Bold indicates the highest mean in each column.

B	P19		MIMIC-III	
	AUROC	**AUPRC**	**AUROC**	**AUPRC**
3	72.17±0.41	17.29±0.36	**89.97±0.32**	66.41±0.30
6	**73.14±0.76**	**17.73±0.37**	89.87±0.27	**66.43±0.32**
12	72.32±1.03	17.42±0.46	89.81±0.19	66.18±0.62

E Scope and Generalization

OPERA is evaluated on retrospective ICU classification tasks using fixed within-cohort training, validation, and test splits. Observation patterns can reflect both patient state and the clinical practices through which measurements are collected. The learned associations therefore characterize predictive monitoring patterns rather than causal effects of measurement.

The three benchmarks establish the utility of this representation across different tasks and variable sets. Generalization across institutions or changes in monitoring practice remains an important extension, requiring external evaluation beyond the current within-cohort experiments.