Warpformer: A Multi-scale Modeling Approach for Irregular Clinical Time Series

Jiawen Zhang*
The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
jzhang302@connect.hkust-gz.edu.cn

Shun Zheng†
Microsoft Research Asia
Beijing, China
shun.zheng@microsoft.com

Wei Cao
Microsoft Research Asia
Beijing, China
wei.cao@microsoft.com

Jiang Bian
Microsoft Research Asia
Beijing, China
jiang.bian@microsoft.com

Jia Li†
The Hong Kong University of Science
and Technology (Guangzhou)
Guangzhou, China
jialee@ust.hk

ABSTRACT

Irregularly sampled multivariate time series are ubiquitous in various fields, particularly in healthcare, and exhibit two key characteristics: intra-series irregularity and inter-series discrepancy. Intra-series irregularity refers to the fact that time-series signals are often recorded at irregular intervals, while inter-series discrepancy refers to the significant variability in sampling rates among diverse series. However, recent advances in irregular time series have primarily focused on addressing intra-series irregularity, overlooking the issue of inter-series discrepancy. To bridge this gap, we present Warpformer, a novel approach that fully considers these two characteristics. In a nutshell, Warpformer has several crucial designs, including a specific input representation that explicitly characterizes both intra-series irregularity and inter-series discrepancy, a warping module that adaptively unifies irregular time series in a given scale, and a customized attention module for representation learning. Additionally, we stack multiple warping and attention modules to learn at different scales, producing multi-scale representations that balance coarse-grained and fine-grained signals for downstream tasks. We conduct extensive experiments on widely used datasets and a new large-scale benchmark built from clinical databases. The results demonstrate the superiority of Warpformer over existing state-of-the-art approaches. [1]

CCS CONCEPTS

• **Applied computing → Health informatics**; • **Computing methodologies → Neural networks**; • **Mathematics of computing → Time series analysis**.

---
*This work was done during the internship at Microsoft Research Asia, Beijing, China.
†Corresponding Author
[1]Our code is available at https://github.com/imJiawen/Warpformer

KEYWORDS

clinical time series, irregularly sampled time series, multi-scale representation

1 INTRODUCTION

With the rapid trend of digitalization in clinical systems, a large accumulation of clinical time-series data has attracted broad research attention, from both computer science and medical communities [14, 21, 54], to explore machine learning solutions over diverse scenarios of digital therapeutics, such as risk stratification and early warning [18, 27, 59], clinical outcome prediction [12, 31], and treatment recommendation [29, 42]. Advanced learning techniques over clinical time series indeed have profound impacts in the real world, such as improving the efficiency and the quality of clinical practices, relieving the burdens of clinicians and nurses, and in some sense, facilitating the equity in terms of the distribution of medical resources [11, 15, 25].

As a specific type of irregularly sampled multivariate time series, clinical time series exhibit two prominent characteristics: *intra-series irregularity* and *inter-series discrepancy*. *Intra-series irregularity* refers to the fact that variate signals within each time series are usually recorded in irregular intervals in practice, breaking up the traditional assumption of regularly sampled time series. Besides, *inter-series discrepancy* denotes the presence of dramatically different sampling rates among multiple time series, leading to a significant imbalance in the occurrence of different signals.

To facilitate understanding, we select some representative time series from MIMIC-III [15], a large-scale healthcare database, and plot them in Figure 1a. We can observe that various vital physiological indicators, such as heart rate, glucose, oxygen saturation, and sodium, are irregularly measured, and some common clinical interventions, such as per os intake and normal saline, are also irregularly issued. In addition to intra-series irregularity, we can also
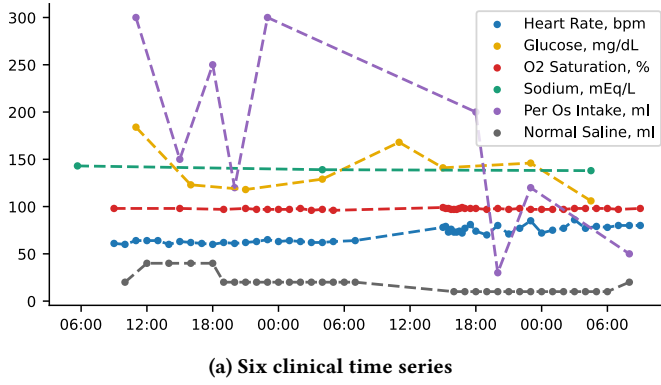
(a) Six clinical time series
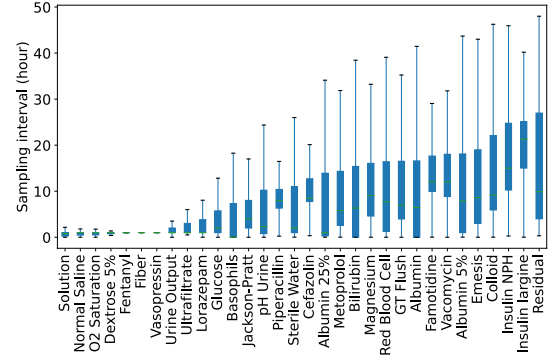
(b) The distribution of sampling intervals for different clinical signals

Figure 1: We select some representative time series from the MIMIC-III [15] database to intuitively illustrate two prominent characteristics, *intra-variate irregularity* and *inter-variate discrepancy*, of irregularly sampled multivariate time series.

observe that heart rate is frequently measured while sodium is sporadically sampled, exhibiting significant discrepancy in sampling frequency. Further, to provide a comprehensive view of inter-series discrepancy, we present the distribution of sampling intervals for tens of typical signals in Figure 1b.

These two data characteristics present substantial challenges in modeling irregularly sampled multivariate time series, particularly in the context of clinical data. First, *intra-series irregularity* prevents the direct adoption of classical time-series models [20, 30, 35, 49, 55, 60, 61] because these models do not include specific mechanisms to handle irregularity in time, which essentially conveys indispensable information on its own [5, 57]. Recent advances in irregular time series have made remarkable progresses to tackle *intra-series irregularity*, such as extending recurrent neural networks [5], leveraging neural ordinary differential equations (ODEs) [6, 32], and introducing time representations [13, 17, 23, 37, 38, 43, 57]. Nevertheless, to the best of our knowledge, these methods do not pay attention to and address the dilemma caused by *inter-series discrepancy* in unifying multivariate time series of significantly different granularities: on one hand, a fine-grained unification retains detailed variations for frequently monitored signals but leads to very sparse placements of sporadically observed signals; on the other hand, a coarse-grained unification provides more balanced placements for all type of signals and can be used to obtain much clearer overall trending, but it sacrifices detailed variations for high-frequency signals.

To bridge this gap, this paper introduces Warpformer, a novel approach that fully considers both *intra-series irregularity* and *inter-series discrepancy*. Warpformer starts with a specific input representation that unifies all signals in the original scale and encodes both signal values and other valuable information brought by *intra-series irregularity*, such as sampling timepoints and intervals. Besides, this representation preserves both temporal and signal dimensions to explicitly present the underlying inter-series discrepancy in sampling frequency. Given this input representation, we stack several Warpformer layers, each of which consists of a warping module

and a doubly self-attention module. The warping module is a specifically designed differentiable network that directly operates on our representation for irregular time series and performs either down-sampling or up-sampling operations to unify all series in a new scale. Following this, the doubly self-attention module, a Transformer [44]-like network, is responsible for representation learning on the unified data representation. As a result, multiple Warpformer layers together produce multi-scale representations for irregular time series, which are added through a residual connection to support downstream tasks.

We validate the effectiveness of Warpformer by comparing it with various state-of-the-art solutions on widely used datasets, such as PhysioNet [39] and Human Activity [32]. Moreover, we also construct a new large-scale benchmark with five critical clinical tasks from the MIMIC-III [15] database to provide more thorough comparisons. Our results indicate that Warpformer, with its specific designs tailored to the characteristics of irregular time series, achieves significant improvements and becomes a new state-of-the-art on these benchmark datasets.

To sum up, our contributions include:

- To the best of our knowledge, this work is the first one that highlights the importance of both *intra-series irregularity* and *inter-series discrepancy* for irregularly sampled multivariate time series. Besides, Warpformer is also the first multi-scale approach for irregular time series.
- We provide extensive experiments to demonstrate the superiority of Warpformer and to verify our critical designs. More importantly, we have constructed a new benchmark with a much larger scale and diversified tasks, which can benefit future research in this field.

2 RELATED WORK

This work focuses on irregularly sampled clinical time series and highlights the multi-scale modeling capability. Besides, we introduce a similar idea of dynamic time warping to enable the adaptive unification for irregular time series. Thus we review related work from these aspects in this section.

Modeling Irregularly Sampled Time Series. Existing methods for modeling irregularly sampled time series have apparent limitations. Early methods [24, 51, 52, 56] used either hourly aggregation or forward imputation to obtain uniformly spaced intervals, overlooking meaningful temporal patterns in irregular sampling. Later approaches, including modifying updating equations in recurrent neural networks [5, 26, 28], learning neural ODEs [6, 32], introducing time-based attentions [37, 38], adding extra time representations [13, 23, 43], and employing graphs to model interactions [57], made remarkable progresses on capturing sampling irregularity but did not seriously consider significant discrepancy in sampling frequency across different series. We note that two emerging paradigms from these approaches are more related to our work: one is to interpolate irregularly observed values to regularly spaced reference points [37, 38]; the other is to unfold irregular time series into a long sequence of "(value, type, time)" tuples [13, 23, 43]. However, the former unifies all time series in one group of pre-specified reference time points, limiting its capability in identifying a data-oriented unification to balance fine-grained and coarse-grained information. Besides, the latter is easy to suffer from the severe imbalance in the observations of different signals and lose the opportunity to capture the general trending hidden from the coarse-grained and balanced view. In contrast, we seriously consider the challenge of *inter-series discrepancy* and equip Warpformer with the multi-scale capability to balance fine-grained and coarse-grained views.

Multi-scale Modeling for Time Series. It is well understood that multi-scale modeling plays a critical role in generic time series. For instance, previous studies [23, 24, 53] on electronic health records have shown the importance of capturing patient states with multiple time granularity. Besides, recent studies [20, 35] have demonstrated that equipping Transformer extensions for time series with the multi-scale capability can bring significant performance improvements. However, these multi-scale approaches only work in the context of regularly spaced time series, and it is non-trivial to adapt them into the setup of irregularly sampling. Accordingly, Warpformer fills this gap and opens up an avenue for multi-scale analysis on irregularly sampled multivariate time series.

Dynamic Time Warping. Existing studies [3, 22, 33] across various application scenarios have witnessed the existence of temporal misalignment. The underlying reasons could be phase shift, sampling rates, precision, etc. To unify these misaligned signals, dynamic time warping (DTW) was proposed [2] and further developed [7, 58]. Traditionally, people use DTW to adjust temporal matching by dynamic programming [2, 3, 7, 33]. In recent years, the need for input-dependent DTW has driven the development of various learning-based alignment solutions, such as temporal transformer nets [22], diffeomorphic temporal alignment nets [48], dynamic temporal pooling [16], and warping based on two-sided distributions [34]. While sharing a similar idea of introducing warping when unifying different signals, this work fundamentally differs from existing DTW-based studies. First, our warping module is the first design that connects differentiable DTW to irregular time series. In contrast, existing studies only consider DTW for regularly sampled time series. Besides, we have formulated unique operations,

which will be specified in Section 3.2, to stimulate the adaptive unification of different signals in a given scale. This functionality is different from existing DTW studies that serve very different purposes, mostly for aligning and clustering. Last, to the best of our knowledge, this is also the first time a warping module is embedded into a neural architecture to support multi-scale learning.

3 METHODOLOGY

In Figure 2, we give an overview of the information flow in Warpformer and highlights our specific designs within the Warpformer layer. In a nutshell, Warpformer starts from a specific input encoder providing rich and structured data representations for irregular time series (Section 3.1) and stacks several Warpformer layers to enable multi-scale representation learning. Each Warpformer layer consists of a warping module (Section 3.2), unifying input representations in a given scale while in the meanwhile preserving the same structured format, and a doubly self-attention module (Section 3.3), responsible for learning high-level representations from unified input data. In this way, multiple Warpformer layers produce multi-scale representations, which are added via a residual connection and then fed into a task decoder (Section 3.4) to support downstream applications.

3.1 Input Encoder

Let $\left\{ [(t_i^k, x_i^k)]_{i=1}^{L^k} \right\}_{k=1}^K$ denote the irregularly sampled multivariate time series of a specific patient, where we have K variates, the k-th variate contains L^k irregularly sampled observations, and the i-th observation of the k-th variate is composed of the recording time t_i^k and the recorded value x_i^k. While holding the advantage of high flexibility in accommodating various signals with variable lengths and sampling rates, this data format does not fit naturally for batch processing in modern neural networks training and inference. Thus, we turn to develop a specific data representation for Warpformer.

First, we collect all unique timestamps by taking an union operation over $[\{t_i^k\}_{i=1}^{L^k}]_{k=1}^K$ and organize them in ascending order as $T \in \mathbb{R}^L$, where L denotes the number of all unique timestamps. Then we fill in a value matrix $X \in \mathbb{R}^{K \times L}$, a type matrix $E \in \mathbb{R}^{K \times L}$, and a mask matrix $M \in \mathbb{R}^{K \times L}$ according to the following rules: $X_{k,j} = x_i^k, E_{k,j} = k, M_{k,j} = 1$ if T_j equals to t_i^k; $X_{k,j} = 0, E_{k,j} = 0, M_{k,j} = 0$ otherwise. Given such a data organization, we apply several encoding functions to obtain the input representation $H \in \mathbb{R}^{K \times L \times D}$ as $H = f^{\text{val}}(X) + f^{\text{type}}(E) + f^{\text{abs}}(T) + f^{\text{rel}}(T, M)$, where f^{val} denotes one fully-connected mapping to transform a single value to an embedding of size D, $f^{\text{type}}(\cdot)$ denotes a lookup table that transform categorical variate indicators into type embeddings (also size D), and $f^{\text{abs}}(\cdot), f^{\text{rel}}(\cdot, \cdot)$ are functions to obtain time-related embeddings. Similar to encoding time-sensitive patterns in previous studies [37, 45], we design $f^{\text{abs}}(\cdot)$, which is in essence a combination of several parallel sinusoidal mappings and one linear mapping, to encode the absolute time. To be specific, the d-th element of $f^{\text{abs}}(\cdot)$'s output embedding is defined as $w_1 \cdot T + b_1$ if d equals to 1 and $\sin(w_d \cdot T + b_d)$ otherwise ($1 < d \le D$). Here $\{w_d, b_d\}_{d=1}^D$ make up the parameters of $f^{\text{abs}}(\cdot)$. Besides, inspired by the success of relative positional encoding in sequence learning [36], we develop
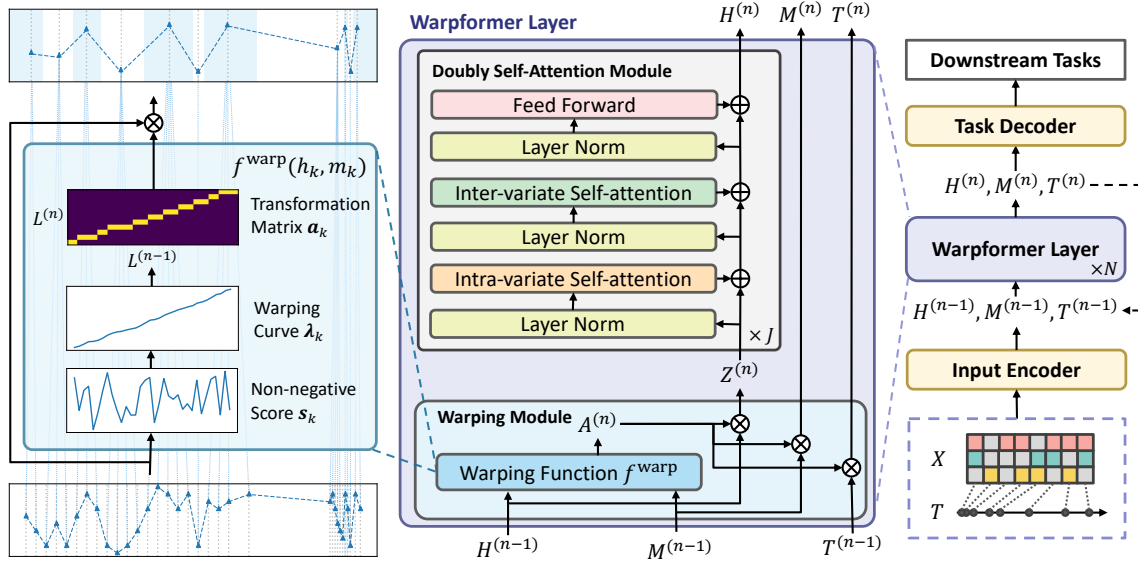
Figure 2: An overview of our Warpformer model: in the right side, we visualize the overall information flow in Warpformer; in the middle, we plot the detailed information flow within a Warpformer layer; in the left side, we visualize crucial intermediate results during the calculation of f^{warp} for the k-th variate to facilitate understanding of how our warping module works.

$f^{\text{rel}}(\cdot, \cdot)$ to encode relative-time information. In this function, we first calculate the sequence of sampling intervals for each variate based on T and M and then apply a two-layer perceptron over the interval value to obtain the relative-time embedding. Last, we feed H together with M and T (broadcasted from $\mathbb{R}^L$ to $\mathbb{R}^{K \times L}$) to subsequent modules.

Connection and Distinction Between Our Input Encoder and Existing Ones. The importance of incorporating distinctive information, such as sampling irregularity, into the learning process for irregularly sampled time series has been widely recognized [5, 6, 37, 57]. Sharing the same spirit, our input representation H encapsulates various critical information, including not only variate values and types but also sampling time and frequencies. But different from recent advancements that built on top of either regularly interpolated representations [37, 38] or a flattened sequence with varying types of signals interleaved [13, 23, 43], our representation retains both temporal and variate dimensions ($\mathbb{R}^{K \times L}$) along with irregular intervals. Such a structured format has several benefits, such as explicitly presenting concurrent observations of certain signals and, more importantly, clearly revealing inter-series discrepancies in sampling frequency.

3.2 Warping Module

The intention of introducing a warping module is to adaptively unify diverse irregularly sampled series into calibrated positions following a specific granularity for further representation learning. Nevertheless, this functionality poses several requirements to the design of this module. First, this module should produce input-dependent unification that fits for the underlying data distribution and benefits downstream tasks. To this end, this module should be differentiable so that supervision signals from downstream tasks

can guide appropriate unification. Besides, this module should support both down sampling and up sampling to align both coarse-grained and fine-grained information. Last, we want to preserve the warping output in the same format as the input $H \in \mathbb{R}^{K \times L \times D}$ owing to the same consideration of explicitly presenting *inter-series discrepancy*, as specified in Section 3.1. With a full consideration of these requirements, we develop our warping module as follows.

An Overview. Let $(H^{(n-1)}, M^{(n-1)}, T^{(n-1)})$ be the input to the warping module of the n-th Warpformer layer ($n \in \{1, \cdots, N\}$), and we use $L^{(n-1)}$ to denote the corresponding sequence length. Our warping module aims to produce the transformation tensor $A^{(n)} \in \mathbb{R}^{K \times L^{(n)} \times L^{(n-1)}}$ based on the encoded representation $H^{(n-1)} \in \mathbb{R}^{K \times L^{(n-1)} \times D}$ and obtain the transformed representation $Z^{(n)} \in \mathbb{R}^{K \times L^{(n)} \times D}$, the new mask $M^{(n)} \in \mathbb{R}^{K \times L^{(n)}}$, and the new anchor time $T^{(n)} \in \mathbb{R}^{K \times L^{(n)}}$ as: $A^{(n)} = f^{\text{warp}}\left(H^{(n-1)}, M^{(n-1)}\right)$, $Z^{(n)} = A^{(n)} \otimes H^{(n-1)}$, $M^{(n)} = A^{(n)} \otimes M^{(n-1)}$, and $T^{(n)} = A^{(n)} \otimes T^{(n-1)}$, where $f^{\text{warp}}(\cdot, \cdot)$ denotes a differentiable function that emits the warping transformation tensor $A^{(n)}$, $\otimes$ refers to a batch matrix product, and $L^{(n)}$ is a hyper-parameter pre-defined as the new sequence length in the n-th layer. Note that, in order to support representation learning on the first layer over the raw granularity, we simply compose K identity mappings into $A^{(1)}$ as $A^{(1)} = [I_1^{L \times L}, \cdots, I_K^{L \times L}]$, thus we have $H^{(1)} = H^{(0)} = H$, $M^{(1)} = M^{(0)} = M$, and $T^{(1)} = T^{(0)} = T$. For subsequent layers that operate in adaptively identified granularity, we adopt a parameterized instantiation of f^{warp}.

Calculating Warping Curves. To facilitate understanding, below we specifically show how to obtain a mapping matrix $a_k = A_{k,:,:}^{(n)} \in \mathbb{R}^{L^{(n)} \times L^{(n-1)}}$ from the enriched representation $h_k = H_{k,:,:}^{(n-1)} \in$

$\mathbb{R}^{L^{(n-1)} \times D}$ and the corresponding mask $\boldsymbol{m}_k = M^{(n-1)}_{k,:} \in \mathbb{R}^{L^{(n-1)}}$ of the k-th variate. First, we compute a non-negative score for each observation entry in $\boldsymbol{h}_k$: $\boldsymbol{s}_k = f^s(\boldsymbol{h}_k) \odot \boldsymbol{m}_k$, where $\boldsymbol{s}_k \in \mathbb{R}^{L^{(n-1)}}$ is the score vector of the k-th variate, $f^s(\cdot)$ is the score function, and $\odot$ denotes the element-wise product. Since each entry in $\boldsymbol{h}_k$ already includes sufficient intra-series and inter-series information owing to the doubly self-attention module of the last layer, we simply instantiate $f^s(\boldsymbol{h}_k)$ as a two-layer fully connected network. In this way, we can obtain variate-specific scores and also take the global context into account. Then we apply a normalized cumulative summation over $\boldsymbol{s}_k$ to obtain the warping curve $\boldsymbol{\lambda}_k \in \mathbb{R}^{L^{(n-1)}}$, so we have $\boldsymbol{\lambda}_{k,i} = (\sum_{i'=1}^{i} \boldsymbol{s}_{k,i'})/(\sum_{i'=1}^{L^{(n-1)}} \boldsymbol{s}_{k,i'})$.

Calculating Transformation Mappings. This warping curve is non-descending, and its values are located within $[0, 1]$. Therefore, by dividing $[0, 1]$ into $L^{(n)}$ segments and using segment boundaries to cut the warping curve $\boldsymbol{\lambda}_k$, we naturally derive a mapping from $\mathbb{R}^{L^{(n-1)}}$ to $\mathbb{R}^{L^{(n)}}$. To be specific, we obtain the left boundaries of $L^{(n)}$ segments dividing $[0, 1]$ as $\boldsymbol{r}^1 = [0, \frac{1}{L^{(n)}}, \cdots, \frac{L^{(n)}-1}{L^{(n)}}]$ and the right boundaries as $\boldsymbol{r}^2 = [\frac{1}{L^{(n)}}, \frac{2}{L^{(n)}}, \cdots, 1]$. To use $\boldsymbol{r}^1, \boldsymbol{r}^2 \in \mathbb{R}^{(n)}$ to cut $\boldsymbol{\lambda}_k \in \mathbb{R}^{(n-1)}$, we first extend the second dimension for $\boldsymbol{r}^1, \boldsymbol{r}^2$ and the first dimension for $\boldsymbol{\lambda}_k$ and then apply broadcast operations along the new dimension to expand them into the same space of $\mathbb{R}^{L^{(n)} \times L^{(n-1)}}$. We denote the resulting matrices of $\boldsymbol{r}^1, \boldsymbol{r}^2$, and $\boldsymbol{\lambda}_k$ as R^1, R^2, and Λ_k, respectively. Note that the operation of $\Lambda_k - R^1 \ge 0$ gives us the mask information indicating whether an element in λ_k is on the right side of the left boundary of a specific segment. Similarly, $R^2 - \Lambda_k \ge 0$ tells us whether an element is on the left side of the right boundary of a given segment. Given the fact that an element belongs to a segment if and only if this element is located in the range covered by the left and right boundaries, we can obtain the matrix a_k^m defining the mapping from $\mathbb{R}^{L^{(n-1)}}$ to $\mathbb{R}^{L^{(n)}}$ as $a_k^m = (\Lambda_k - R^1 \ge 0)$ & $(R^2 - \Lambda_k \ge 0)$, where & denotes the logical And operation.

Enabling Up-sampling. Using the segment boundaries specified by $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ to cut the warping curve naturally supports the down-sampling behavior, which happens when several entries in $\boldsymbol{\lambda}_k$ fall into the same segment. But this scheme cannot assign any entry in $\boldsymbol{\lambda}_k$ to multiple positions in $\mathbb{R}^{L^{(n)}}$ if we want a fine-grained unification when $L^{(n)} > L^{(n-1)}$. To support up-sampling, our overall idea is to traverse the leftmost non-zero boundary in the upper triangle matrix $(\Lambda_k - R^1 \ge 0) \odot \Lambda_k$ and the rightmost non-zero boundary in the lower triangle $(R^2 - \Lambda_k \ge 0) \odot \Lambda_k$ and then copy these two boundary curves, essentially triggering up-sampling behaviors when $L^{(n)} > L^{(n-1)}$, to supplement the initial mapping matrix a_k^m. In practice, we can implement this idea very efficiently by adjusting R^1 and R^2. First, we can use the values in the rightmost non-zero boundary of $(R^2 - \Lambda_k \ge 0) \odot \Lambda_k$ to update $\boldsymbol{r}^1$ as $\boldsymbol{r}_u^1 = \min(\boldsymbol{r}^1, \text{perRowMax}((R^2 - \Lambda_k \ge 0) \odot \Lambda_k))$ and then broadcast $\boldsymbol{r}_u^1 \in \mathbb{R}^{L^{(n)}}$ along a new dimension to obtain $R_u^1 \in \mathbb{R}^{L^{(n)} \times L^{(n-1)}}$. The adjusted R_u^1 ensures that the mappings recorded by the rightmost non-zero boundary of $(R^2 - \Lambda_k \ge 0)$ is definitely included in the non-zero entries of $(\Lambda_k - R_u^1 \ge 0)$. And we can perform a similar

adjustment to obtain R_u^2. Finally, we calculate a_k^m with new R_u^1 and R_u^2 as $a_k^m = (\Lambda_k - R_u^1 \ge 0)$ & $(R_u^2 - \Lambda_k \ge 0)$.

Enabling Differentiation. Moreover, we note that the operation involved in obtaining a_k^m is non-differentiable. Accordingly, we utilize the following equation to compute a mapping matrix that preserves the flow of gradient computations: $a_k^u = a_k^m \odot (\max(\Lambda_k - R^1, 0) + \max(R^2 - \Lambda_k, 0))$. Note that the value of a non-zero element in a_k^u denotes the summation of its distance to the left boundary and the distance to the right, which exactly equals to the segment width, namely $\frac{1}{L^{(n)}}$. This fact means that a_k^u is a mapping matrix with uniform weights. Furthermore, to stabilize the magnitude of the transformed embedding in $Z^{(n)}$, which should be irrelevant to the number of raw data points assigned to a specific segment, we normalize a_k^u by its per-row summation as $a_k = a_k^u / \text{rowSum}(a_k^u)$.

It is apparent that the above operations to obtain a_k can be easily parallelized for all K variates, and all these operations constitute the function $f^{\text{warp}}(H^{(n-1)}, M^{(n-1)})$ to produce the final transformation tensor $A^{(n)}$. Moreover, we can observe that $T^{(n-1)}$ does not participate in the computation of $A^{(n)}$ because $H^{(n-1)}$ already includes time-related information in the input encoder. We preserve the calculation from $T^{(n-1)}$ to $T^{(n)}$ mainly for visualization and interpretation purposes.

3.3 Doubly Self-attention Module

To perform effective learning over $Z^{(n)}$, the representation produced by the warping module in the n-th layer, a desired learning module on top of $Z^{(n)}$ should not only hold sufficient capacities to effectively capture predictive patterns but also retain the representation structure that includes both time and variate dimensions, in other words, keep operating in the space of $\mathbb{R}^{K \times L^{(n)} \times D}$, so as to preserve the inductive bias of explicitly presenting "concurrent" observations and to support further processing of subsequent warping modules. Although Transformer [44], an extraordinarily successful neural architecture applied in various sequence learning scenarios [4, 8, 9, 19, 40], appears to be a straightforward option, its self-attention mechanism, assuming the input sequence lies in the space of $\mathbb{R}^{L^{(n)} \times D}$, does not support our target of capturing both intra-series and inter-series patterns. Therefore, we develop a novel doubly self-attention mechanism as a customized extension of the Transformer encoder to fit our representation structure.

Specifically, we perform two consecutive standard self-attention operations on two views of the organized representation $Z^{(n)} \in \mathbb{R}^{K \times L^{(n)} \times D}$. In the first view, we regard $Z^{(n)}$ as K sequences of embeddings with each sequence in $\mathbb{R}^{L^{(n)} \times D}$. As for the second, we rearrange $Z^{(n)}$ into $L^{(n)}$ sequences of embeddings with each one in $\mathbb{R}^{K \times D}$. Besides, each self-attention operation also follows prior arts [46] to perform layer normalization [1] over the input in advance and include a residual connection. Such a doubly self-attention mechanism enables the information exchange across both intra-series and inter-series dimensions. After that, we finish one-layer encoding by applying the vanilla position-wise feed-forward mapping in Transformer, which naturally fits our representation

format. Last, to enable effective learning and facilitate sufficient exchanges of both intra-series and inter-series information, we stack multiple such encoding layers to compose the doubly self-attention module that maps $Z^{(n)}$ into $H^{(n)}$ for the subsequent process. Note that $Z^{(n)}$ and $H^{(n)}$ are in the same space as $\mathbb{R}^{K \times L^{(n)} \times D}$, and we do not need to change $M^{(n)}$ and $T^{(n)}$.

3.4 Task Decoder

With multi-layer stacking of Warpformer layers, each of which consists of a warping module and a doubly self-attention module, we can naturally obtain multi-scale representations by aggregating the output of each layer, denoted as $\{(H^{(n)}, M^{(n)})\}_{n=1}^{N}$. While these representations correspond to different sequence lengths, denoted as $\{L^{(n)}\}_{n=1}^{N}$, we adopt attention-based pooling operations to obtain fix-sized embeddings for downstream tasks. Note that these representations correspond to different sequence lengths, denoted as $\{L^{(n)}\}_{n=1}^{N}$, due to the layer-by-layer warping transformations towards more coarse-grained views. To leverage these multi-scale representations in variable shapes, we condense the time and variate dimensions of each $H^{(n)}$ to obtain a fixed-size embedding in $\mathbb{R}^{D}$. To be specific, we leverage an attention-based aggregation operation: $f^{\mathrm{agg}}(\text{Query}, \text{Key}, \text{Value}) = \text{Softmax}(\text{Query} \cdot \text{Key}^{T}) \cdot \text{Value}$, where $\text{Query} \in \mathbb{R}^{D}$ is a query vector, $\text{Key} \in \mathbb{R}^{L \times D}$ is a key matrix, $\text{Value} \in \mathbb{R}^{L \times D}$ is a value matrix, and we use $\cdot$ to denote vector-matrix product. Note that no matter how L varies, the output of f^{agg} is still in $\mathbb{R}^{D}$. To condense the time dimension, we simply set $\text{Query} = Q^{t}$, $\text{Key} = \text{Tanh}(\text{Linear}(\boldsymbol{h}_{k}))$, $\text{Value} = \boldsymbol{h}_{k}$, $\boldsymbol{u}_{k} = f^{\mathrm{agg}}(\cdot, \cdot, \cdot)$, where $Q^{t} \in \mathbb{R}^{D}$ denotes a parameter, $\boldsymbol{h}_{k} \in \left\{ H_{k,:,:}^{(n)} \in \mathbb{R}^{L^{(n)} \times D} \right\}_{k=1}^{K}$ refers to the k-th slicing of $H^{(n)}$ along the variate dimension, and similarly $\boldsymbol{u}_{k} \in \mathbb{R}^{D}$ denotes the k-th slicing of $U^{(n)} \in \mathbb{R}^{K \times D}$. Furthermore, we set $\text{Query} = Q^{v}$, $\text{Key} = \text{Tanh}(\text{Linear}(U^{(n)}))$, $\text{Value} = U^{(n)}$, and $\boldsymbol{v}^{(n)} = f^{\mathrm{agg}}(\cdot, \cdot, \cdot)$ to remove the variate dimension. In this way, we transform the tensor $H^{(n)}$ into the fixed-size embedding $\boldsymbol{v}^{(n)} \in \mathbb{R}^{D}$. By applying the same operations with shared parameters over all $\{H^{(n)}\}_{n=1}^{N}$, we obtain a bunch of equal-size embeddings $\{\boldsymbol{v}^{(n)}\}_{n=1}^{N}$ encapsulating multi-scale patterns. We add these embeddings into the final representaiton $\boldsymbol{v} = \sum_{n=1}^{N} \boldsymbol{v}^{(n)}$ to support downstream tasks. For example, given a multi-class classification task with C classes in total, we can perform the following computation: $\hat{\boldsymbol{y}} = \text{Softmax}(W^{y} \boldsymbol{v} + \boldsymbol{b}^{y})$, where the prediction vector $\hat{\boldsymbol{y}} \in \mathbb{R}^{C}$ includes a predicted probability distribution over C classes, and $W^{y} \in \mathbb{R}^{C \times D}, \boldsymbol{b}^{y} \in \mathbb{R}^{C}$ are task-specific parameters.

4 EXPERIMENTS

In this section, we provide extensive experiments to demonstrate the effectiveness of Warpformer.

4.1 Experimental Settings

4.1.1 Datasets. We follow existing studies [5, 6, 37] to compare different models on the *PhysioNet* dataset [39] and the *Human Activity* dataset [32]. The *PhysioNet* dataset comprises 4,000 instances and focuses on predicting in-hospital mortality. It includes 4 types of demographics and 37 physiological signals collected during the initial 48 hours of ICU admission. The median length of instances

in this dataset is 72. The *Human Activity* dataset aims to classify specific human activities among seven types for each timepoint in a segment of irregularly sampled time series. It consists of 6,554 time-series segments with a total of 12 channels. All instances in this dataset have a fixed length of 50 timepoints. We mainly follow the existing setup [37] to divide the original dataset into the train, validation, and test sets, except that we do not shuffle instances in the *Human Activity* dataset, which are obtained by truncating five long sequences, to avoid the potential information leakage. Nevertheless, these two datasets cover very limited clinical prediction tasks domain, signal types, and data instances, which may lead to biased estimates in comparing different approaches. Meanwhile, we note that the *MIMIC-III* database [15] contains many clinical scenarios calling for accurate predictions over diversified clinical signals, most of which are irregularly sampled. Then we refer to [25, 43, 47] and construct a new benchmark with five representative clinical tasks, 103 clinical signals (61 biomarkers and 42 interventions), and hundreds of thousands of instances. The specific clinical tasks include in-hospital <u>mor</u>tality (MOR), <u>dec</u>ompensation (DEC), <u>l</u>ength <u>o</u>f <u>s</u>tay (LOS), next timepoint <u>w</u>ill <u>b</u>e <u>m</u>easured (WBM), and <u>c</u>linical <u>i</u>ntervention <u>p</u>rediction (CIP), each of which serves as a new dataset for evaluation. Additional details on these datasets can be found in Appendix A.

4.1.2 Metrics. In line with previous studies, we evaluate the performance using the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) for the *PhysioNet* dataset and the five datasets derived from *MIMIC-III*. Besides, following the prior art, we introduce Accuracy as the evaluation metric for the *Human Activity* dataset. For multi-class or multi-label classifications, we calculate AUROC and AUPRC scores for each individual class (label) and then compute the average as the dataset-level score. To eliminate the randomness, we conduct each experiment with five random seeds and report both the mean and standard deviation of the results.

4.1.3 Baselines. We organize existing methods applicable to irregular clinical time series into five paradigms. The first is to introduce irregularity-sensitive updating mechanisms into recurrent neural networks, such as **RNN-Mean**, **RNN-Forward**, **RNN-Δ_t**, **RNN-Decay**, and **GRU-D** in [5] as well as **Phased-LSTM** [28]. The second is to model the hidden continuous dynamics behind irregular observations via neural ODEs, such as **ODE-RNN** and **L-ODE-ODE** [32]. Since neural ODEs are computational intensive, the third paradigm, including **IP-Net** [38] and **mTAND** [37], attempts to introduce continuous-time representations and align irregular observations into regularly spaced reference points to fit for classical time-series models. The fourth paradigm tackles the time irregularity by organizing multiple types of irregular observations into a long sequence of "(time, type, value)" tuples and modeling their interactions via self-attention mechanisms, such as **SeFT** [13], or via graph neural networks, such as **RainDrop** [57]. Last, the fifth paradigm refers to recent domain-specific methods for electronic health records (EHRs). Typical examples include **AdaCare** [24], a multi-scale model but operating on regularly aggregated time series, and **STraTS** [43], a similar variant of **SeFT** introducing trainable time encodings and auxiliary learning objectives. We include

Table 1: Evaluation results (mean ± std %) on *PhysioNet* and *Human Activity*.

Model	PhysioNet		Human Activity	
	AUROC	AUPRC	Accuracy	AUPRC
RNN-Mean	55.5 ± 8.7	20.3 ± 7.5	74.9 ± 1.4	65.5 ± 2.4
RNN-Forward	84.2 ± 0.7	52.6 ± 0.7	76.7 ± 0.6	68.7 ± 0.6
RNN-Δ_t	65.7 ± 3.2	25.2 ± 1.5	74.6 ± 0.6	65.3 ± 2.0
RNN-Decay	49.9 ± 0.9	15.7 ± 0.4	75.9 ± 1.3	64.4 ± 1.5
GRU-D	84.7 ± 0.3	52.6 ± 0.3	75.0 ± 1.0	64.3 ± 1.7
Phased-LSTM	78.1 ± 0.5	40.7 ± 3.6	73.5 ± 1.7	60.8 ± 1.0
ODE-RNN	83.7 ± 0.6	53.3 ± 1.1	76.0 ± 1.2	64.9 ± 2.6
L-ODE-ODE	81.2 ± 1.3	46.6 ± 3.6	76.7 ± 1.4	77.1 ± 1.4
SeFT	70.8 ± 0.3	29.9 ± 0.9	75.8 ± 3.3	66.1 ± 3.3
RainDrop	69.2 ± 4.2	28.4 ± 3.6	70.2 ± 1.6	62.3 ± 2.6
IP-Nets	75.6 ± 1.1	37.5 ± 2.3	73.5 ± 1.7	64.7 ± 2.4
mTAND	86.0 ± 0.4	54.6 ± 0.9	81.3 ± 0.3	73.5 ± 0.8
Warpformer	**86.6 ± 0.6**	**56.7 ± 0.7**	**84.9 ± 0.7**	**81.1 ± 0.9**

these two EHR-specific methods in the experiments on large-scale datasets built from *MIMIC-III*.

4.1.4 ***Implementation Details***. We employ slightly different hyperparameter configurations across the datasets in our experiments. For a comprehensive reference, we have summarized the hyperparameter settings used for each dataset in Appendix B.2. Our model is implemented using PyTorch 1.9.0, and both training and inference are performed on CUDA 11.3. We utilize the Adam optimizer with an initial learning rate of 10^{-3}. The models are trained for a maximum of 50 epochs, and early stopping is applied if there is no improvement on the validation set for 5 consecutive training epochs. All experiments are conducted on NVIDIA Tesla V100 GPUs. To ensure fair comparisons with other baselines, we explore the hyperparameter search space around their best-reported configurations.

To accommodate the variability in input sequence lengths across datasets, we introduce a normalized length $\tilde{L} = L/L_{data}$, where L_{data} is the median length of training instances. When $\tilde{L}$ is less than 1, the warping layer predominantly downsamples the time series, while a value greater than 1 signifies upsampling. For example, in the case of the *Human Activity* dataset with a median length of 50, if $\tilde{L}^{(n)} = 0.2$, it means that the new sequence length in the n-th layer is 10 (i.e., $10/50 = 0.2$). The median lengths for each clinical dataset can be found in Table 4. We utilized specific scales for different datasets in our main experimental results and ablation tests. For the *PhysioNet* dataset, we use three scales with $\tilde{L}^{(1)} = 0.2$ and $\tilde{L}^{(2)} = 1$. For the *Human Activity* dataset, we use three scales with $\tilde{L}^{(1)} = 1.2$ and $\tilde{L}^{(2)} = 1$. Lastly, for the *MIMIC-III* based datasets, we utilize two scales with $\tilde{L}^{(1)} = 1$. Given the importance of $\tilde{L}^{(n)}$ as a predefined hyperparameter in our model, we provide a sensitivity analysis in Section 4.2.3 and Appendix B.3 to assess its impact and determine the optimal combinations.

4.2 Experimental Results

4.2.1 ***Main Results***. We include the overall comparison results on *PhysioNet* and *Human Activity* in Table 1. Besides, we present

AUROC and AUPRC results on *MIMIC-III*-based datasets in Table 2. In general, we observe that different baselines present apparent performance variations across different datasets and evaluation metrics. Interestingly, certain advanced methods, such as mTAND, demonstrate excellent performance on the *PhysioNet* and *Human Activity* datasets. However, despite extensive hyper-parameter tuning, these methods do not exhibit comparable performance on the *MIMIC-III*-based datasets. On the other hand, basic extensions of recurrent neural networks, such as RNN-Mean and RNN-Decay, perform poorly on *PhysioNet* but surprisingly achieve competitive and robust results on *MOR*, *DEC*, and *LOS* tasks. We conjecture the underlying reason is related to the matching degree between the effective patterns of a specific dataset and the unique assumptions held by a specific model. Nevertheless, with the capability of encoding various crucial information in irregular time series and unifying both fine-grained and coarse-grained information, Warpformer achieves remarkable performance improvements over the most competitive baselines in all setups.

Additionally, we observed that the improvements achieved by Warpformer vary across tasks, with specific tasks, e.g., WBM, benefiting more than others, e.g., MOR. One possible explanation for this variation is the inherent difficulty of tasks. The WBM task involves predicting a substantial number of biomarkers to be measured in the upcoming hour, necessitating the model to capture inter-variable discrepancies. In contrast, the MOR task only involves binary classification, which is relatively straightforward and demands less attention to inter-variable discrepancies. The increased complexity and inter-variable challenges in the WBM task likely contribute to the significant gains achieved by the Warpformer model compared to the MOR task. Another factor that can impact performance is the length of observation windows for each task. Shorter observation windows may limit the model's ability to capture complex temporal patterns and obscure intra- and inter-variable discrepancies. For example, the WBM task requires a 48-hour look-back window, necessitating the model to capture intricate temporal patterns. In contrast, the CIP task only requires a 6-hour look-back window, which could explain its minor improvement compared to the tasks with longer observation windows.

4.2.2 ***Ablation Tests***. We conducted ablation tests on Warpformer to illustrate the significance of critical designs. Table 3 includes the results of three warping-related variants, 1) disabling the up-sampling functionality (No Up-sampling); 2) using the identical mapping to substitute the warping-based mapping (Identical Mapping); 3) using hourly aggregation to substitute the warping module (Hourly Aggregation). The results demonstrate that the warping module plays an important role compared to identical mapping. Notably, the ability to upsample significantly impacts the performance, as the absence of upsampling functionality can even lead to inferior results compared to identical mapping. These findings underscore the necessity of capturing finer-grained details of sparsely sampled signals during multi-scale modeling. Furthermore, the experimental results on the *PhysioNet* dataset reveal that inappropriate schemes for obtaining coarse-grained representations, such as aggregating all data within a time slot, can yield worse results than a model without a multi-scale setting. These observations validate the effectiveness of the adaptive unification strategy employed.

Table 2: AUROC and AUPRC (mean ± std %) of different methods on five datasets built from *MIMIC-III*.

Model	MOR		DEC		LOS		WBM		CIP	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
RNN-Mean	89.5 ± 0.3	62.9 ± 0.7	98.4 ± 0.2	83.6 ± 0.8	76.7 ± 0.1	31.2 ± 0.1	79.4 ± 0.3	25.8 ± 0.6	86.9 ± 0.3	45.6 ± 0.2
RNN-Forward	88.5 ± 0.3	60.4 ± 1.2	97.1 ± 0.6	75.6 ± 2.2	76.4 ± 0.2	30.8 ± 0.1	74.6 ± 4.9	22.4 ± 3.0	86.4 ± 0.4	45.2 ± 0.3
RNN-Δ_t	87.4 ± 1.0	55.8 ± 2.4	97.3 ± 0.5	74.5 ± 0.8	76.2 ± 0.2	30.6 ± 0.2	72.5 ± 4.2	20.4 ± 2.1	50.6 ± 0.6	25.2 ± 0.2
RNN-Decay	89.1 ± 0.4	62.6 ± 1.0	98.4 ± 0.3	84.2 ± 2.4	77.0 ± 0.1	31.6 ± 0.1	77.8 ± 2.2	24.5 ± 1.9	76.2 ± 9.6	39.8 ± 4.6
GRU-D	87.6 ± 0.5	59.9 ± 0.6	97.6 ± 1.0	77.9 ± 5.6	76.5 ± 0.1	30.9 ± 0.3	74.4 ± 6.9	22.7 ± 4.9	84.6 ± 1.8	43.3 ± 1.7
Phased-LSTM	86.7 ± 0.3	53.7 ± 0.6	97.5 ± 0.1	78.2 ± 0.4	75.6 ± 0.2	30.2 ± 0.1	77.4 ± 0.3	24.3 ± 0.4	84.8 ± 0.2	44.0 ± 0.1
SeFT	88.0 ± 0.4	59.3 ± 1.4	98.5 ± 0.2	86.7 ± 0.7	76.1 ± 0.1	31.0 ± 0.1	83.8 ± 0.4	33.1 ± 0.3	86.6 ± 0.2	45.0 ± 0.1
RainDrop	87.6 ± 0.1	59.1 ± 0.4	98.0 ± 0.2	82.7 ± 0.5	76.0 ± 0.2	30.6 ± 0.4	80.4 ± 0.4	27.7 ± 0.6	86.9 ± 0.4	44.8 ± 0.2
IP-Nets	88.9 ± 0.3	61.9 ± 0.9	98.3 ± 0.1	85.2 ± 0.6	74.3 ± 3.6	28.8 ± 3.7	81.6 ± 0.1	28.1 ± 0.6	85.2 ± 0.6	44.0 ± 0.6
mTAND	89.0 ± 0.2	61.8 ± 0.7	97.2 ± 0.3	74.5 ± 3.2	73.8 ± 0.4	28.3 ± 0.4	66.1 ± 0.2	16.7 ± 0.1	84.2 ± 0.3	42.1 ± 0.2
AdaCare	76.0 ± 0.5	47.7 ± 0.7	94.9 ± 2.0	68.5 ± 9.0	64.9 ± 0.6	22.1 ± 1.2	51.2 ± 0.2	12.9 ± 0.0	64.4 ± 2.4	36.2 ± 1.9
STraTS	89.3 ± 0.1	61.3 ± 0.3	98.6 ± 0.1	84.0 ± 1.6	76.6 ± 0.2	31.4 ± 0.3	79.1 ± 2.1	26.2 ± 2.4	87.8 ± 0.2	45.6 ± 0.2
Warpformer	**90.3 ± 0.1**	**64.6 ± 0.4**	**99.0 ± 0.1**	**90.0 ± 0.4**	**77.7 ± 0.2**	**32.5 ± 0.2**	**85.5 ± 0.1**	**35.5 ± 0.3**	**88.0 ± 0.2**	**46.4 ± 0.2**

Table 3: Ablation tests of Warpformer on *PhysioNet* and *Human Activity*.

Warpformer	PhysioNet		Human Activity	
	AUROC	AUPRC	Accuracy	AUPRC
Full	**86.6 ± 0.6**	**56.7 ± 0.7**	**84.9 ± 0.7**	**81.1 ± 0.9**
No Up-sampling	85.4 ± 1.2	54.5 ± 1.8	84.4 ± 0.8	79.0 ± 0.7
Identical Map.	84.8 ± 0.4	52.0 ± 2.1	84.8 ± 1.2	80.0 ± 1.4
Hourly Agg.	84.2 ± 0.5	51.6 ± 2.4	N/A	N/A
$-f^{abs}(\cdot)$	83.2 ± 1.3	50.9 ± 1.6	80.6 ± 1.9	74.0 ± 2.3
$-f^{rel}(\cdot)$	86.3 ± 1.1	54.5 ± 2.7	84.8 ± 0.7	80.5 ± 1.0
$-f^{type}(\cdot)$	86.3 ± 0.9	55.7 ± 2.3	84.3 ± 0.4	79.3 ± 0.9
Input Pooling	47.3 ± 0.1	14.8 ± 0.0	32.4 ± 0.0	15.7 ± 0.1

To further explore the importance of different components in the input encoding process, we conducted additional ablation tests, and the results are also presented in Table 3. The notation $-f^{abs}(\cdot)$, $-f^{rel}(\cdot)$, and $-f^{type}(\cdot)$ represent the removal of absolute-time embeddings, relative-time embeddings, and type embeddings, respectively. The term "Input pooling" refers to the application of a simple average pooling operation that collapses the temporal and variate dimensions. The results demonstrate that the model experiences a significant performance drop when any encoding components are missing. Particularly, the absence of absolute-time encoding has the most pronounced impact on the model's performance. This finding is intuitive as $f^{abs}(\cdot)$ captures the absolute position of each sample within a continuous time period, which is crucial for effectively modeling irregularly sampled time series. These ablation tests provide strong evidence supporting the importance of each encoding component in the input representation of Warpformer. By considering these components together, Warpformer is able to capture the intricate temporal relationships and characteristics of irregularly sampled time series data.

*4.2.3 **Multi-scale Effects**.* We study the multi-scale effects of Warpformer with different setups of the number of Warpformer layers N, i.e., the number of scales, and the hyper-parameters

$\{\tilde{L}^{(n)}\}_{n=1}^{N}$, i.e., the unification granularity. Figure 3 includes the results on *PhysioNet* and *Human Activity*, where we show the performance as a function of $\tilde{L}^{(n)}$. Note that in the case of the *Human Activity* dataset, hidden states must be generated for each time step since it involves per-step prediction. Consequently, the 2 scales mentioned in Figure 3b actually correspond to 3 Warpformer layers ($N = 2$), where $\tilde{L}^{(2)} = 1$. Similarly, the 3 scales mentioned in Figure 3d correspond to 4 Warpformer layers ($N = 3$).

These findings suggest that the optimal number of layers applied varies across different datasets. The optimal number is often not very large, indicating that using 2 or 3 scales in Warpformer for encoding is sufficient to generate highly effective representations for downstream tasks. We observed that increasing the number of layers, which involves repeating up- or down-sampling operations, did not yield substantial additional benefits. In fact, it was noticed that excessive stacking of scales could potentially weaken the patterns learned in the previous layers. This observation emphasizes the importance of striking the right balance in the number of scales used, avoiding unnecessary complexity that could impede the model's ability to capture and leverage meaningful information.

Moreover, the results highlight the significance of selecting an appropriate unification granularity in Warpformer. Interestingly, different datasets showed distinct preferences for down-sampling and up-sampling configurations. To be specific, extreme down-sampling had the most substantial performance improvement on the *PhysioNet* dataset, whereas the *Human Activity* dataset preferred up-sampling configurations. We speculate that the disparity in performance can be attributed to the nature of the prediction tasks in each dataset. The *PhysioNet* dataset involves a single prediction target for the entire series, prioritizing the capture of long-term trends and overall patterns. Thus, a more coarse-grained representation proves beneficial in this context. On the other hand, the *Human Activity* dataset requires accurate predictions at each time point, necessitating fine-grained information to discern subtle variations and dependencies within the temporal sequences. Therefore, the up-sampling configurations are particularly advantageous for
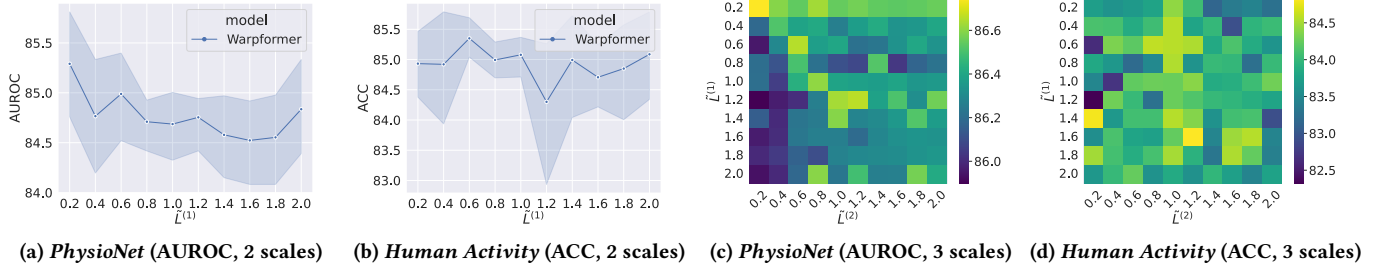
(a) *PhysioNet* (AUROC, 2 scales) (b) *Human Activity* (ACC, 2 scales) (c) *PhysioNet* (AUROC, 3 scales) (d) *Human Activity* (ACC, 3 scales)

Figure 3: The multi-scale effects of Warpformer on *PhysioNet* and *Human Activity*.
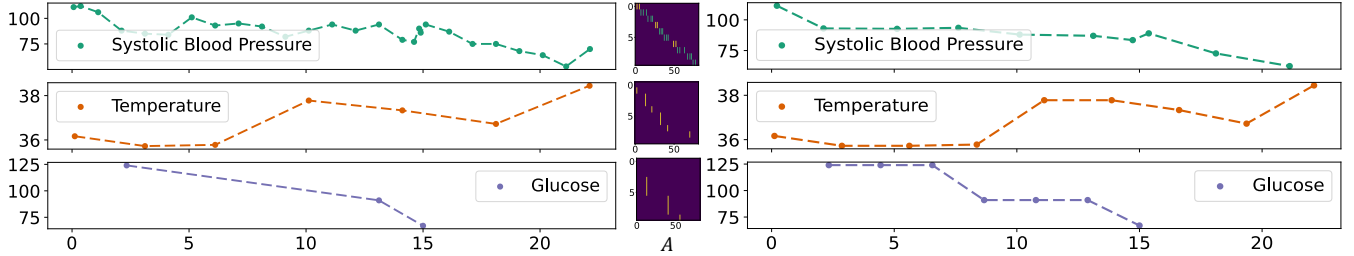

Figure 4: Visualization of the input time series (left) and its adaptive unification (right), as well as the learned corresponding alignment matrix A (middle). The length of the y-axis of A is $\tilde{L}^{(n)}$, i.e., the size of the produced unification, and the x-axis is $\tilde{L}^{(n-1)}$, i.e., the length of the input time series.

this dataset. These findings shed light on the importance of adapting the unification granularity to the specific characteristics of the dataset and prediction task. By tailoring the level of down-sampling or up-sampling, Warpformer can effectively capture the relevant temporal patterns and optimize its performance accordingly.

4.3 Case Study

To gain deeper insights into how the warping mechanism enhances predictions in downstream tasks, we conducted a comprehensive case study focusing on instances that exhibited substantial improvements after the warping process. The visual analysis, as presented in Figure 4, includes three representative physiological signals from the MIMIC-III-based datasets, showcasing the original input signals, the corresponding learned alignment matrix A, and the output series after adaptive unification.

The visualizations demonstrate that the warped signals faithfully capture the trends present in the original signals, and the warping module exhibits the ability to compute variate-specific alignment schemas. For densely sampled signals, such as systolic blood pressure, the warping module downsamples the original series while preserving irregular patterns and important local fluctuations. This is achieved by retaining the intra-variate irregularity during the unification process, which cannot be achieved through traditional hourly aggregation. Such capability is advantageous in multi-scale modeling as it prevents the loss of irregular time patterns at coarse-grained levels and retains valuable details. On the other hand, for sparsely sampled signals, e.g., body temperature and glucose, the warping module generates additional interpolated sample points, resulting in a more fine-grained representation of the coarse-grained

signal. These interpolated sample points provide more detailed features, facilitating the model's ability to capture subtle variations in the temporal sequences.

The case study clearly demonstrates that the warping mechanism successfully preserves intra-variate irregularities and alleviates inter-variate discrepancies. This underscores the crucial role of the adaptive unification process in Warpformer, which significantly contributes to the model's performance and ability to capture the underlying dynamics of the irregular clinical time series.

5 CONCLUSION AND FUTURE WORK

In conclusion, our paper has highlighted the importance of addressing both intra-series irregularity and inter-series discrepancy when dealing with irregularly sampled multivariate time series, particularly in clinical scenarios. We have presented the first multi-scale approach, Warpformer, which has achieved remarkable performance across various datasets. However, it is important to acknowledge that our work has several limitations at this stage. Firstly, the reliance on pre-specified hyper-parameters, such as the number of layers and per-layer unification granularity, poses a challenge in terms of efficiency in hyper-parameter tuning. Secondly, the maintenance of both temporal and variate dimensions in internal feature representations introduces additional space complexity. Finally, Warpformer's dependence on gradients from downstream tasks limits its use to supervised settings, which may affect its generalization performance in real-world scenarios with limited labeled data. Future work will explore ways to address these limitations and improve the efficiency and generalization of our approach.

REFERENCES

[1] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *ArXiv* abs/1607.06450 (2016).

[2] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*.

[3] Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. 2020. (k, l)-Medians Clustering of Trajectories Using Continuous Dynamic Time Warping. In *SIGSPATIAL*.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *NeurIPS 2020*.

[5] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 1–12.

[6] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural Ordinary Differential Equations. In *NeurIPS*. 6572–6583.

[7] Marco Cuturi and Mathieu Blondel. 2017. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*. PMLR, 894–903.

[8] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

[9] Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *ICASSP*. IEEE, 5884–5888.

[10] Marzyeh Ghassemi, Mike Wu, Michael C. Hughes, Peter Szolovits, and Finale Doshi-Velez. 2017. Predicting intervention onset in the ICU with switching state space models. *AMIA Summits on Translational Science Proceedings* 2017 (2017), 82 – 91.

[11] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific data* 6, 1 (2019), 1–18.

[12] JoonNyung Heo, Jihoon G Yoon, Hyungjong Park, Young Dae Kim, Hyo Suk Nam, and Ji Hoe Heo. 2019. Machine learning–based model for prediction of outcomes in acute stroke. *Stroke* 50, 5 (2019), 1263–1265.

[13] Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten M. Borgwardt. 2020. Set Functions for Time Series. In *ICML*, Vol. 119. 4353–4363.

[14] Peter B Jensen, Lars J Jensen, and Søren Brunak. 2012. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics* 13, 6 (2012), 395–405.

[15] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9.

[16] Dongha Lee, Seonghyeon Lee, and Hwanjo Yu. 2021. Learnable dynamic temporal pooling for time series classification. In *AAAI*.

[17] Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. 2019. Predicting Path Failure In Time-Evolving Graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[18] Jia Li, Yu Rong, Helen Meng, Zhihui Lu, Timothy Kwok, and Hong Cheng. 2018. TATC: predicting Alzheimer's disease with actigraphy data. In *SIGKDD*. 509–518.

[19] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2018. Neural Speech Synthesis with Transformer Network. In *AAAI*.

[20] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. 2022. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *ICLR*.

[21] Yang Liu, Yu Rong, Zhuoning Guo, Nuo Chen, Tingyang Xu, Fugee Tsung, and Jia Li. 2023. Human Mobility Modeling During the COVID-19 Pandemic via Deep Graph Diffusion Infomax. In *AAAI*.

[22] Suhas Lohit, Qiao Wang, and Pavan Turaga. 2019. Temporal transformer networks: Joint learning of invariant and discriminative time warping. In *CVPR*. 12426–12435.

[23] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. 2020. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *SIGKDD*. 647–656.

[24] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020. AdaCare: Explainable Clinical Health Status Representation Learning via Scale-Adaptive Feature Extraction and Recalibration. In *AAAI*. 825–832.

[25] Matthew B. A. McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. 2021. A comprehensive EHR timeseries pre-training benchmark. In *CHIL*. 257–278.

[26] Michael C. Mozer, Denis Kazakov, and Robert V. Lindsey. 2017. Discrete Event, Continuous Time RNNs. *CoRR* abs/1710.04110 (2017). arXiv:1710.04110 http://arxiv.org/abs/1710.04110

[27] Sankavi Muralitharan, Walter Nelson, Shuang Di, Michael McGillion, PJ Devereaux, Neil Grant Barr, and Jeremy Petch. 2021. Machine Learning–Based Early Warning Systems for Clinical Deterioration: Systematic Scoping Review. *Journal of medical Internet research* 23, 2 (2021), e25187.

[28] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences. In *NueIPS*. 3882–3890.

[29] Milad Zafar Nezhad, Najibesadat Sadati, Kai Yang, and Dongxiao Zhu. 2019. A deep active survival analysis approach for precision treatment recommendations: Application of prostate cancer. *Expert Systems with Applications* 115 (2019), 16–26.

[30] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.

[31] Zhaozhi Qian, Yao Zhang, Ioana Bica, Angela Wood, and Mihaela van der Schaar. 2021. SyncTwin: Treatment Effect Estimation with Longitudinal Outcomes. *NeurIPS* 34 (2021).

[32] Yulia Rubanova, Tian Qi Chen, and David Duvenaud. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In *NeurIPS*. 5321–5331.

[33] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), 159–165.

[34] Erik Scharwächter, Jonathan Lennartz, and Emmanuel Müller. 2021. Differentiable Segmentation of Sequences. In *ICLR*.

[35] Mohammad Amin Shabani, Amir H. Abdi, Lili Meng, and Tristan Sylvain. 2023. Scaleformer: Iterative Multi-scale Refining Transformers for Time Series Forecasting. In *ICLR*.

[36] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *ACL*.

[37] Satya Narayan Shukla and Benjamin Marlin. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *ICLR*.

[38] Satya Narayan Shukla and Benjamin M. Marlin. 2019. Interpolation-Prediction Networks for Irregularly Sampled Time Series. In *ICLR*.

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

[40] David So, Quoc Le, and Chen Liang. 2019. The evolved transformer. In *ICML*. PMLR, 5877–5886.

[41] Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. 2017. Clinical intervention prediction and understanding with deep neural networks. In *Machine Learning for Healthcare Conference*. 322–337.

[42] Panagiotis Symeonidis, Christos Andras, and Markus Zanker. 2021. Treatment Recommendations for COVID-19 Patients along with Robust Explanations. In *CBMS*. 207–212.

[43] SINDHU TIPIRNENI and CHANDAN K REDDY. 2022. Self-Supervised Transformer for Sparse and Irregularly Sampled Multivariate Clinical Time-Series. *ACM Trans. Knowl. Discov. Data* 1, 1 (2022).

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[45] Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. 2021. On Position Embeddings in BERT. In *ICLR*.

[46] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning Deep Transformer Models for Machine Translation. In *ACL*.

[47] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. 2020. MIMIC-Extract: a data extraction, preprocessing, and representation pipeline for MIMIC-III. In *CHIL*. 222–235.

[48] Ron Shapira Weber, Matan Eyal, Nicki Skafte Detlefsen, Oren Shriki, and Oren Freifeld. 2019. Diffeomorphic Temporal Alignment Nets. In *Neural Information Processing Systems*.

[49] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *NeurIPS*.

[50] Mike Wu, Marzyeh Ghassemi, Mengling Feng, Leo A Celi, Peter Szolovits, and Finale Doshi-Velez. 2017. Understanding vasopressor intervention and weaning: risk prediction in a public heterogeneous clinical time series database. *Journal of the American Medical Informatics Association* 24, 3 (2017), 488–495.

[51] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. 2018. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *SIGKDD*. 2565–2573.

[52] Zhen Xu, David R. So, and Andrew M. Dai. 2021. MUFASA: Multimodal Fusion Architecture Search for Electronic Health Records. In *AAAI*. 10532–10540.

[53] Muchao Ye, Junyu Luo, Cao Xiao, and Fenglong Ma. 2020. Lsan: Modeling long-term dependencies and short-term correlations with hierarchical attention for risk prediction. In *CIKM*. 1753–1762.

[54] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. 2018. Artificial intelligence in healthcare. *Nature biomedical engineering* 2, 10 (2018), 719–731.

[55] Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. 2022. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186* (2022).

[56] Xianli Zhang, Buyue Qian, Yang Li, Yang Liu, Xi Chen, Chong Guan, and Chen Li. 2021. Learning Robust Patient Representations from Multi-modal Electronic Health Records: A Supervised Deep Learning Approach. In *SDM*. 585–593.

[57] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Graph-Guided Network For Irregularly Sampled Multivariate Time Series. In *ICLR*.

[58] Zheng Zhang, Ping Tang, and Thomas Corpetti. 2020. Time adaptive optimal transport: A framework of time series similarity measure. *IEEE Access* 8 (2020), 149764–149774.

[59] Le Zheng, Oliver Wang, Shiying Hao, Chengyin Ye, Modi Liu, Minjie Xia, Alex N Sabo, Liliana Markovic, Frank Stearns, Laura Kanov, et al. 2020. Development of an early-warning system for high-risk patients for suicide attempt using deep learning and electronic health records. *Translational psychiatry* 10, 1 (2020), 1–10.

[60] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*.

[61] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*. 27268–27286.
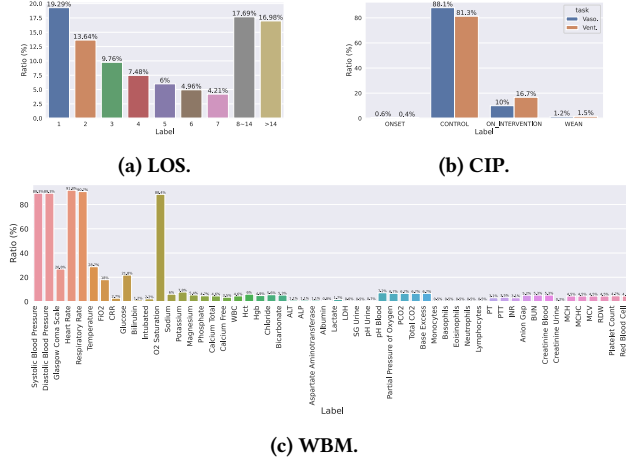
A MIMIC-III-BASED DATASETS

(a) LOS.

(b) CIP.

(c) WBM.

Figure 5: Label distribution of LOS, CIP, and WBM datasets.

MIMIC-III database [15] contains hospital admission records of 53, 423 de-identified patients. We selected 61 common biomarker variables and 42 widely used interventions in intensive care units (ICU) [43]. We also follow the previous studies [38, 47] to ensure the sufficient duration of hospital stay by removing the records with the length of stay less than 48 hours. Figure 7 illustrates the distribution of sampling intervals in the in-hospital mortality task for all clinical signals, visually representing the wide range of variate and significant inter-variate discrepancies in the clinical time series. To imitate the practical situation, we sort the clinical records according to the time of admission, where the training set includes the earliest 80% of admission records. We equally divide the rest into validation and testing sets. There is no admission overlap in different sets to avoid data leakage issues. Similar to [25, 38, 47], we select a diverse set of tasks to cover different clinical scenarios. Table 4 summarizes the statistics of these tasks, which will be briefly introduced one by one in the following.

In-hospital Mortality (MOR) The goal of the MOR task is to predict whether a patient will decease at the end of this hospital stay. In this task, models can observe only the first 48 hours, which is advantageous from a practical view since the sooner clinicians identify the risks, the more prompt they can implement interventions. As performed in existing studies [37, 38, 47], we define this task as a binary classification problem, where 1 indicates that the patient deceased in this hospital stay, otherwise 0. The overall in-hospital mortality rate of this benchmark is 11.66%.

Decompensation (DEC) The objective of DEC is to determine whether a patient will decease in the next 24 hours based on the data within a 12-hour time window. This task, also referred to as imminent mortality prediction, has been studied widely as a signal for more general physiological decompensation [11, 25]. We formulate this task as a binary classification as well, but sampling data throughout the entire hospital stay in the rolling form. In this way, we obtained 311, 161 samples with a mortality rate of 1.81%.

Length Of Stay (LOS) Compared with other tasks, the LOS prediction provides a more fine-grained view of patients' physiological

states, helping clinicians better monitor the progress of the disease. As did in [51], we formulate this task as a classification problem with 9 classes, where class 1-7 corresponds to 1-7 days of stay, respectively, class 8 for more than eight days but less than two weeks of stay, and class 9 for living over two weeks. As a rolling task, its observation is sampled every 12 hours throughout the entire stay, where the sliding window size is 24 hours. The distribution of labels for this dataset can be found in Figure 5a.

Next Timepoint Will Be Measured (WBM) The WBM task is a multi-label classification problem with the goal of predicting which biomarker signals will be measured in the next hour [25]. This task is beneficial to assist clinicians in developing the subsequent treatment plan. To fulfill this task, the model needs to make predictions for 54 biomarkers, each of which is a binary classification problem. If a signal was measured within the following hour, it is assigned a label of 1, otherwise 0. The WBM task is conducted over the entire stay of a patient, where the sampling stride is 12 hours and the observation window size is 48 hours. We provide the distributions of each prediction target in Figure 5c.

Clinical Intervention Prediction (CIP) As a critical clinical application, accurately predicting the clinical interventions can largely release the burden of clinicians in practice [10, 50]. Similar to [41, 47], we also formulate the CIP task as a multi-class classification problem. This task consists of two subtasks in terms of different intervention types: *mechanical ventilation* and *vasopressor*. The prediction target for each type should be one of the four possible options: 1) onset, 2) wean, 3) continuing on the intervention, and 4) continuing to stay off the intervention. For more details on the distribution of prediction targets, please refer to Figure 5b. The input of this task is the historical data within the 6-hour lookback window, and the prediction target is the state of the intervention within the next 4-hour lookahead window. The CIP task is also conducted over the entire stay of a patient, where the sampling stride is 6 hours.

B EXPERIMENTS

B.1 Performance Metrics

Clinical downstream tasks, e.g., mortality prediction, frequently encounter significant data imbalance issues. Consequently, we employed AUROC and AUPRC as the primary evaluation metrics. Due to the balanced nature of the *Human Activity* dataset and the
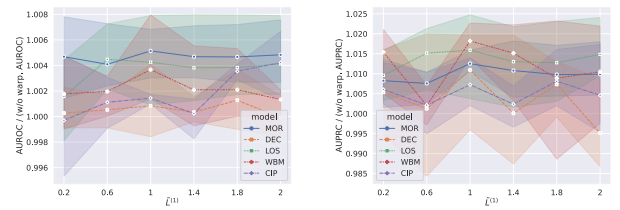

(a) MIMIC-III (AUROC, 2 scales) (b) MIMIC-III (AUPRC, 2 scales)

Figure 6: Performance of Warpformer with varying $\tilde{L}^{(1)}$ on MIMIC-III-based datasets.

Table 4: Specifications for five clinical datasets derived from the *MIMIC-III* database (*BC*: binary classification, *ML*: multi-label classification, *MC*: multi-class classification).

Task (Abbr.)	Type	# Train	# Val.	# Test	Median Seq. Len.	Clinical Scenario
In-hospital Mortality (MOR)	BC	39, 449	4, 939	4, 970	63	Early warning
Decompensation (DEC)	BC	249, 045	31, 896	30, 220	34	Outcome prediction
Length Of Stay (LOS)	MC	249, 572	31, 970	30, 283	34	Outcome prediction
Next Timepoint Will Be Measured (WBM)	ML	223, 867	28, 754	27, 038	78	Treatment recommendation
Clinical Intervention Prediction (CIP)	MC	223, 913	28, 069	27, 285	10	Treatment recommendation
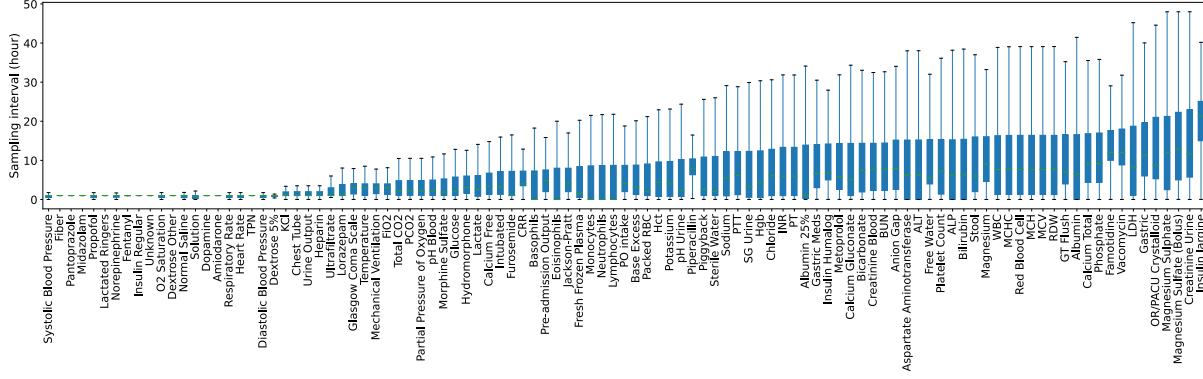
Figure 7: The distribution of sampling intervals for all the clinical signals in the MOR task.

Table 5: Specifications for hyper-parameters.

Dataset	D	Batch size	# Head	# Layers J
PhysioNet	32	32	1	2
Human Activity	64	64	8	3
MIMIC-III	32	32	1	2

prevailing usage of Accuracy in previous studies [37, 38], we also adopted Accuracy for this particular dataset. For each dataset, we select the model parameters that yield the highest AUROC value on the validation set and apply them for evaluation on the test set.

B.2 Further Details on Hyper-parameters

For each dataset, we tailor the dimensionality of hidden states, batch size, and the number of heads and layers in the doubly attention module. Please refer to Table 5 for specific configuration details. In our implementation, each attention head is set to have a dimension of 8. The score function $f^s(\cdot)$ is implemented as a two-layer fully connected network with a ReLU activation function. Through thorough testing, we find that using the Sigmoid function to compute the non-negative score ensures better stability in our model.

B.3 Further analyses on multi-scale hyper-parameters

To further demonstrate the impact of scale numbers (N) and unification granularity ($\tilde{L}^{(n)}$), we provide the performance of two-layer Warpformer on MIMIC-III-based datasets (Figure 6) and the AUROC

of a four-layer Warpformer on the PhysioNet dataset (Figure 8). It shows that MIMIC-III-based datasets consistently yield the best results when $\tilde{L}^{(1)} = 1$. Additionally, the results of the four-scale experiment on the PhysioNet dataset do not surpass the performance of the 3-scale settings. Notably, extreme downsampling applied to $\tilde{L}^{(1)}$ and $\tilde{L}^{(2)}$ achieves relatively strong performance, aligning with our observations in Section 4.2.3.
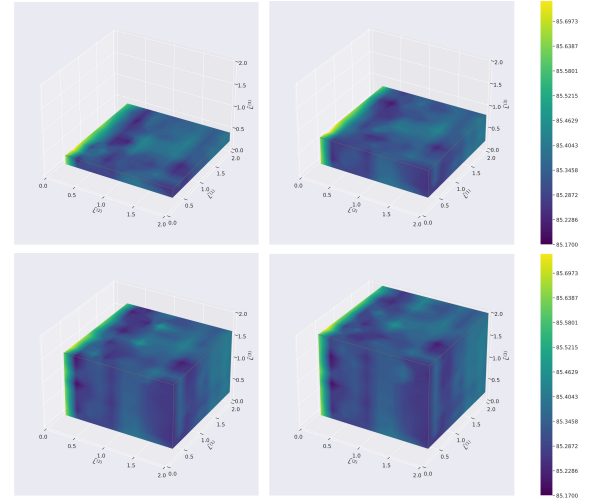
Figure 8: AUROC of the four-layer Warpformer on the PhysioNet dataset with different $\tilde{L}^{(n)}$ settings.