MATCHING SCALES TO PATTERNS: SOFT-SEGMENTED MULTI-SCALE EXPERTS FOR IRREGULAR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Irregular multivariate time series (IMTS) are pervasive in real-world scenarios, characterized by uneven sampling intervals and misaligned observations across variables. Their temporal patterns can vary substantially across samples and periods in aspects such as fluctuation dynamics and changes in sampling density. Distinct aspects of these patterns can be captured at multiple scales, while their variation across samples and evolution over time may call for different scale combinations. However, existing multi-scale IMTS forecasting methods typically apply a fixed scale configuration to all samples and time periods, limiting their ability to adapt scales to varying temporal patterns. To address this limitation, we propose SoftScale, an adaptive soft-segmented multi-scale experts network for IMTS forecasting. SoftScale first employs data-dependent time cursors to construct sample-adaptive soft segments from raw IMTS. For each soft segment, a multi-scale experts network then captures temporal and inter-variable dependencies at each scale. To integrate these dependencies, an MoE router adaptively selects the most suitable scale combination for each soft segment and combines the corresponding scale-specific representations. In this way, SoftScale enables adaptive multi-scale modeling across different samples and periods, leading to more accurate IMTS forecasting. Experiments on four real-world datasets demonstrate that SoftScale outperforms existing state-of-the-art baselines. Our code is available at https://anonymous.4open.science/r/SoftScale/.

1 INTRODUCTION

Time series are prevalent in fields such as healthcare (Lan et al., 2024) and weather forecasting (Verma et al., 2024). Due to sensor failures and sampling constraints, real-world multivariate time series are often irregularly sampled. These series are commonly referred to as irregular multivariate time series (IMTS) (Shukla & Marlin, 2020). IMTS exhibit diverse temporal patterns across samples and periods, such as short-term fluctuations, seasonal variations, and long-term trends, as well as uneven sampling intervals within each variable and misalignment across variables. Previous work has shown that multi-scale modeling is important for capturing such patterns because different scales may capture distinct aspects of temporal patterns (Wang et al., 2024; Cai et al., 2024). Accordingly, different scale combinations may be useful for modeling temporal patterns that vary across samples and over time. However, most multi-scale methods are designed for regular time series (Chen et al., 2024) and therefore cannot be directly applied to IMTS. Existing multi-scale methods for IMTS (Luo et al., 2025; Zhang et al., 2023; Li et al., 2026) apply a fixed scale configuration to all samples and periods, limiting their ability to adapt scales to varying temporal patterns.

Figure 1 provides an illustrative example of diverse temporal patterns in IMTS. For example, during the first 30 hours, V_1 is densely observed and remains relatively stable in Sample 1, whereas it is observed at a moderate rate and follows a different trajectory in Sample 2. V_2 also follows different trajectories across the two samples and is much more densely observed in Sample 1 than in Sample 2. These cross-sample differences in trajectories and sampling densities suggest that different samples may benefit from different scale combinations for representing sample-specific dynamics.

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.

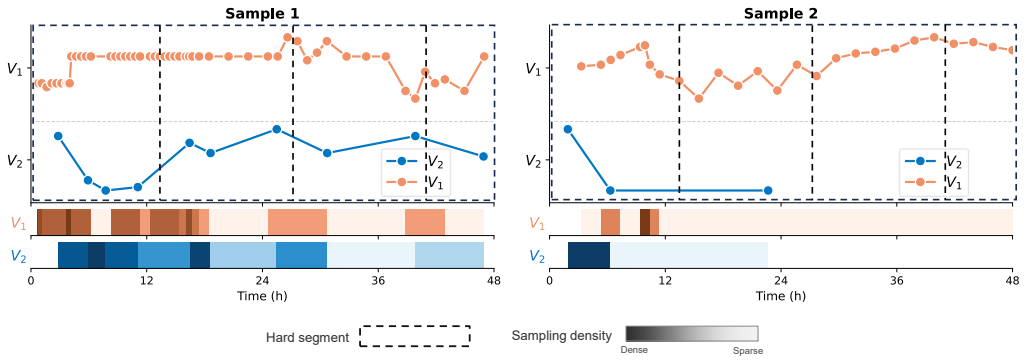


Figure 1: Observations of two variables from two real ICU patient samples in the PhysioNet dataset (Silva et al., 2012). The two variables V_1 and V_2 represent heart rate (HR) and pH, respectively. **Top:** observations of V_1 and V_2 from the two samples. **Bottom:** sampling-density strips for each variable, where strip width represents the interval between adjacent observations and color intensity (dark to light) represents sampling density (dense to sparse). The dashed lines mark illustrative hard segment boundaries.

Beyond cross-sample variation, temporal patterns in IMTS may also change across periods within the same sample. In clinical monitoring, for example, both a patient's condition and observation frequency can change over time (Shukla & Marlin, 2020). Focusing on the temporal evolution within Sample 1 of Figure 1, V_1 is relatively stable and densely observed in the early period but becomes more variable and sparsely observed later; by contrast, V_2 exhibits a comparatively consistent trajectory and sampling profile throughout the same period. These within-sample changes across periods suggest that different scale combinations may be useful for representing period-specific dynamics. For example, densely observed periods can support both fine-grained representations that preserve local fluctuations and coarser representations that emphasize long-term trends, whereas coarse aggregation may discard the limited information in sparsely observed periods (Luo et al., 2025). The differences shown in Figure 1 therefore suggest that scale preferences should be adapted to different samples and periods.

However, existing methods use either raw sequence observations (Zhang et al., 2023) or hard segments (Luo et al., 2025; Li et al., 2026) as their modeling units and apply a fixed scale configuration across periods, limiting their ability to accommodate period-specific scale preferences. Hard segmentation uses fixed or learned boundaries to assign each timestamp to a single segment. This becomes problematic when aspects of temporal patterns, such as fluctuation dynamics and sampling density, change gradually rather than abruptly. The dashed lines in Figure 1 illustrate this issue: observations near a boundary are assigned to only one segment, even if they are similar to observations on the other side of the boundary. Because transition points vary across samples and variables, hard segmentation is sensitive to boundary placement.

Motivated by these limitations, we propose SoftScale, an adaptive soft-segmented multi-scale network for IMTS forecasting. SoftScale consists of three stages. First, the Adaptive Cursor-based Soft Segment (ACSS) module uses the input timestamps to generate data-dependent time cursors and construct sample-adaptive soft segments. Second, the multi-scale experts use the observed values to extract scale-specific temporal representations for each soft segment and model inter-variable dependencies at each scale. Finally, a Mixture-of-Experts (MoE) router adaptively combines the scale-specific representations for each soft segment. Together, soft segmentation adapts to varying patterns across samples and periods. It also guides the MoE router to select suitable scales, allowing SoftScale to learn sample- and period-specific scale preferences.

The main contributions of this paper are summarized as follows:

- We propose SoftScale, an adaptive soft-segmented multi-scale framework for IMTS forecasting that learns sample- and period-specific scale preferences.

- We develop the Adaptive Cursor-based Soft Segment (ACSS) module, which uses input timestamps to generate data-dependent time cursors and construct sample-adaptive soft segments that capture local variations in observation timing.
- Experiments on four benchmark IMTS datasets show that SoftScale outperforms state-of-the-art baselines in most settings while maintaining competitive computational efficiency.

2 Related work

Irregular Multivariate Time Series Forecasting. Existing methods for IMTS forecasting can be divided into two categories based on whether interpolation is applied. The first category employs kernel-based approaches (Shukla & Marlin, 2019; Wu et al., 2021) and Gaussian process-based approaches (Tan et al., 2021; Fortuin et al., 2020; Shukla & Marlin, 2022) to interpolate the raw data, thereby making it regular and easier to process. However, interpolation may obscure the original irregular sampling structure and thereby lose information encoded in the observation times (Luo et al., 2024). In contrast, the second category learns directly from IMTS without any interpolation. ODE-based methods (Rubanova et al., 2019; De Brouwer et al., 2019; Biloš et al., 2021; Kidger et al., 2020) typically incur high computational complexity. GRU-D (Che et al., 2018) enhances RNNs by using time intervals. Recent studies employ graph neural networks (Li et al., 2017; Wu et al., 2020; Zhang et al., 2020) or attention mechanisms (Liu et al., 2024; Zhang & Yan, 2023; Lin et al., 2025) to capture inter-variable and temporal dependencies. However, most existing methods model IMTS at a single scale, limiting their ability to capture distinct aspects of temporal patterns at multiple scales.

Multi-scale Modeling in Time Series. Researchers have found that multi-scale information is important for time series modeling (Ma et al., 2020; Chen et al., 2021; Cheng et al., 2023). Pyraformer (Liu et al., 2022) employs pyramid attention for multi-scale feature extraction, Scaleformer (Shabani et al., 2023) iteratively refines forecasts across temporal scales, and Pathformer (Chen et al., 2024) proposes an adaptive path approach to fuse information across different scales. However, these methods are all designed for regular time series data and cannot effectively adapt to IMTS data. Warpformer (Zhang et al., 2023) is the first to propose a multi-scale modeling method for IMTS, using a warping module that unifies irregular time series at a given scale. Hi-Patch (Luo et al., 2025) and ReIMTS (Li et al., 2026) employ hierarchical frameworks for multi-scale modeling. Nevertheless, these methods apply a fixed scale configuration to all samples and periods, limiting their ability to adapt scales to varying temporal patterns.

Mixture-of-Experts in Time Series. Mixture of Experts (MoE) typically distributes the knowledge learned by the model across multiple expert subnetworks, with a learnable router coordinating their contributions (Shazeer et al., 2017; Zhou et al., 2022). This mechanism has recently been used more frequently to enhance the efficiency of time series foundation models (Shi et al., 2025; Liu et al., 2025a). SoftShape (Liu et al., 2025b) also employs this approach for modeling subsequences of varying shapes, enhancing the ability to distinguish between subsequences and improving the performance of temporal classification. Unlike the aforementioned methods, we use an MoE router to learn sample- and period-specific scale preferences for IMTS.

3 Problem definition

In this study, we focus on the problem of IMTS forecasting. We consider a dataset $\mathcal{D} = \{\mathcal{S}^{(i)}\}_{i=1}^{n}$ consisting of n samples, where i denotes the sample index. Each sample $\mathcal{S}^{(i)}$ is represented as a set of $M^{(i)}$ asynchronous observations: $\mathcal{S}^{(i)} = \{(t_j^{(i)}, z_j^{(i)}, v_j^{(i)})\}_{j=1}^{M^{(i)}}$, where j denotes the observation-event index within the sample, $t_j^{(i)} \in \mathbb{R}^+$ is the timestamp, $z_j^{(i)} \in \mathbb{R}$ is the observed value, and $v_j^{(i)} \in \{1, \ldots, V\}$ is the variable indicator among V total variables. Each sample $\mathcal{S}^{(i)}$ is partitioned by a reference time $t_S^{(i)}$ into a historical window $\mathcal{X}^{(i)} = \{(t_j^{(i)}, z_j^{(i)}, v_j^{(i)}) \in \mathcal{S}^{(i)} \mid j = 1, ..., M^{(i)}, t_j^{(i)} \leq t_S^{(i)}\}$ and a future forecasting window $\mathcal{Y}^{(i)} = \{(t_j^{(i)}, z_j^{(i)}, v_j^{(i)}) \in \mathcal{S}^{(i)} \mid j = 1, ..., M^{(i)}, t_j^{(i)} > t_S^{(i)}\}$. The set of forecast queries $\mathcal{Q}^{(i)} = \{q_j^{(i)}\}_{j=1}^{M_Q^{(i)}}$ is derived from the timestamps and variable indicators in the forecasting window, i.e., $q_j^{(i)} = (t_j^{(i)}, v_j^{(i)})$. Our objective is to learn a forecasting model $\mathcal{F}(\cdot)$

that accurately predicts the corresponding values $\mathcal{Z}^{(i)} = \{z_j^{(i)}\}_{j=1}^{M_Q^{(i)}}$ given the historical observations $\mathcal{X}^{(i)}$ and the target queries $\mathcal{Q}^{(i)}$:

$$\mathcal{F}(\mathcal{X}^{(i)}, \mathcal{Q}^{(i)}) \rightarrow \hat{\mathcal{Z}}^{(i)}, \tag{1}$$

where $\hat{\mathcal{Z}}^{(i)}$ is the set of predicted values approximating the ground truth $\mathcal{Z}^{(i)}$.

4 METHODOLOGY

We describe our framework using a representative sample $\mathcal{S}^{(i)}$ containing T historical observations. For notational simplicity, we omit the sample superscript (i) below. The section is organized as follows: Section 4.1 introduces the Adaptive Cursor-based Soft Segment (ACSS) module; Section 4.2 presents the Multi-scale Experts Network, including scale-specific temporal attention, inter-variable attention, and expert fusion through gated routing; Section 4.3 describes our forecasting approach; and Section 4.4 defines the training loss functions. The overall architecture is illustrated in Figure 2.

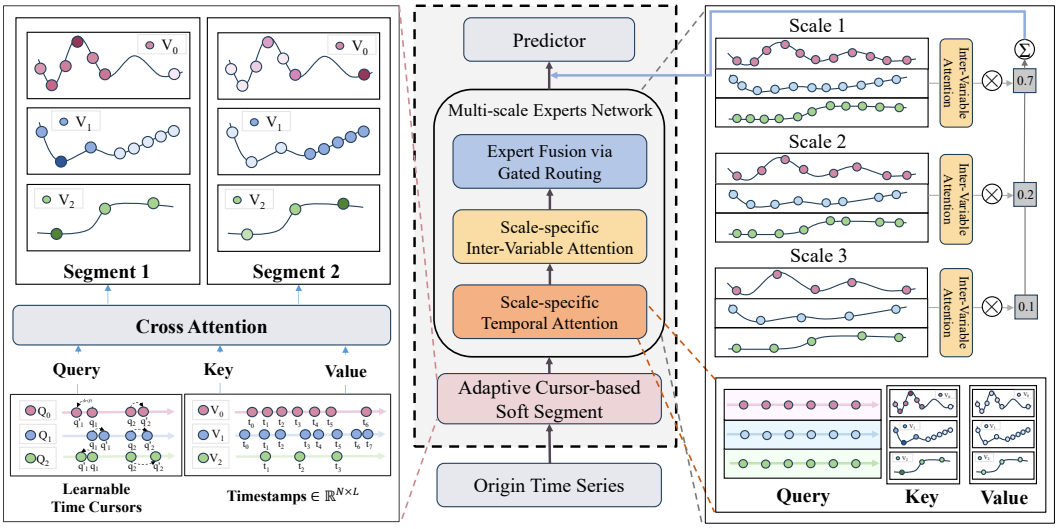


Figure 2: The overall framework of SoftScale. The example input is an IMTS consisting of three variables. First, the Adaptive Cursor-based Soft Segment (ACSS) module uses learnable time cursors to construct sample-adaptive soft segments from the input IMTS. In Segments 1 and 2, the color intensity of each point indicates its assignment weight to the corresponding soft segment (darker indicates a higher weight). Subsequently, the Multi-scale Experts Network captures temporal and inter-variable dependencies at each scale. Then, an MoE router adaptively combines the scale-specific representations for each segment. Finally, an MLP-based predictor generates the final forecasts.

4.1 ADAPTIVE CURSOR-BASED SOFT SEGMENT

To construct sample-adaptive soft segments, we introduce time cursors whose positions are adjusted based on the timestamps of each input IMTS. The resulting soft segments support period-specific multi-scale modeling of temporal patterns. For the input IMTS $\mathcal{S}^{(i)}$, we align timestamps along the temporal dimension to form $\mathcal{T} \in \mathbb{R}^{V \times T}$, zero-padding unobserved entries. We place time cursors along the temporal axis. Each cursor attends to nearby timestamps and weights their relative importance, yielding a soft-segment temporal representation that captures local sampling patterns for subsequent multi-scale modeling.

Initially, for a given number of cursors P, we initialize a time cursor matrix $Q_{\text{init}} \in \mathbb{R}^{V \times P}$ using a uniform distribution, allocating P time cursors to each variable. To generate data-dependent time cursors, we use an MLP f_{drift} to obtain the sample-specific drift $Q_{\text{drift}} = f_{\text{drift}}(\mathcal{T}) \in \mathbb{R}^{V \times P}$. The

MLP consists of two linear layers with a Tanh activation. The final time cursors are computed as follows:

$$Q_{\text{adapt}} = Q_{\text{init}} + \lambda Q_{\text{drift}}, \tag{2}$$

where λ is a hyperparameter controlling the contribution of Q_{drift}.

Following Shukla & Marlin (2021), we define a D-dimensional continuous-time embedding $\phi(\cdot)$ for a scalar time coordinate t:

$$\phi(t)[d] = \begin{cases} \omega_0 \cdot t + \alpha_0, & d = 0, \\ \sin(\omega_d \cdot t + \alpha_d), & 0 < d < D, \end{cases}, \tag{3}$$

where ω_d and α_d are learnable parameters. We apply $\phi(\cdot)$ element-wise to Q_{adapt} and $\mathcal{T}$, and project the resulting embeddings using an MLP f_{time} to obtain the time-cursor queries Q_c, timestamp keys K_c, and timestamp values V_c. Specifically, $Q_c = f_{\text{time}}(\phi(Q_{\text{adapt}})) \in \mathbb{R}^{V \times P \times D}$, while $K_c, V_c = f_{\text{time}}(\phi(\mathcal{T})) \in \mathbb{R}^{V \times T \times D}$. The MLP f_{time} consists of two linear layers with a Tanh activation.

Using cosine similarity, we compute the soft-segment assignment matrix A as:

$$A = \text{Softmax}\left(\frac{Q_c K_c^T}{\|Q_c\| \|K_c\|}\right), \tag{4}$$

where $\|\cdot\|$ denotes the L_2 norm.

We apply A to the timestamp embeddings V_c along the feature dimension to obtain the soft-segment temporal representations $C \in \mathbb{R}^{V \times P \times T \times D}$:

$$C = A \odot V_c. \tag{5}$$

For each time cursor p, the corresponding soft-segment representation in C captures local variations in observation timing. The resulting representation serves as a key for scale-specific temporal attention and a routing signal for learning period-specific scale preferences.

To maintain cursor distinctiveness and prevent mode collapse during training, we introduce a diversity loss that penalizes proximity between cursors via an RBF kernel. For the time cursors $Q_{\text{adapt}} = \{(q_{v1}, q_{v2}, \ldots, q_{vP})\}_{v=1}^{V} \in \mathbb{R}^{V \times P}$, the loss is defined as:

$$\mathcal{L}_{\text{dist}} = \frac{1}{V} \sum_{v=1}^{V} \sum_{1 \le p < k \le P} \exp\left(-\frac{|q_{vp} - q_{vk}|^2}{2\sigma^2}\right), \tag{6}$$

where σ is a hyperparameter that controls the range of interaction between cursors.

4.2 Multi-scale experts network

We first construct the keys for scale-specific temporal attention from the P soft-segment temporal representations introduced in Section 4.1. Then, the two attention modules capture temporal and inter-variable dependencies at each scale. Finally, an MoE router adaptively selects and combines the multi-scale features of each segment to produce the final representations.

4.2.1 Scale-specific temporal attention

Given a predefined scale pool $\mathcal{R} = \{r_s\}_{s=1}^{k}$ containing k scales, r_s denotes the sequence length at the s-th scale. A larger r_s retains finer-grained temporal information, whereas a smaller r_s performs stronger aggregation and produces a coarser representation.

For each scale s, we introduce a learnable scale query $Q_{\text{scale}} \in \mathbb{R}^{V \times r_s \times D}$. We use an MLP f_{scale} to obtain the scale queries $Q_s = f_{\text{scale}}(Q_{\text{scale}}) \in \mathbb{R}^{V \times r_s \times D}$ and keys $K_s = f_{\text{scale}}(C) \in \mathbb{R}^{V \times P \times T \times D}$, where C is defined in Eq. (5). The MLP f_{scale} has the same architecture as f_{time}.

Furthermore, the observed values $\mathcal{U} = \{(v_j, z_j)\}_{j=1}^{T}$ of $\mathcal{S}^{(i)}$ are utilized as the value $V_s \in \mathbb{R}^{V \times T}$. By applying the cross-attention mechanism (Vaswani et al., 2017) as defined in Eq. (7), we derive the temporal representation $E_{\text{scale}} \in \mathbb{R}^{V \times P \times r_s}$ for each specific scale:

$$E_{\text{scale}} = \text{Softmax}\left(Q_s K_s^T\right) V_s \in \mathbb{R}^{V \times P \times r_s}. \tag{7}$$

The resulting E_{scale} captures the temporal patterns of each soft segment at scale s for subsequent inter-variable modeling.

4.2.2 SCALE-SPECIFIC INTER-VARIABLE ATTENTION

To model inter-variable dependencies at each scale, we use a learnable variable embedding $E_{\text{var}} \in \mathbb{R}^{V \times D}$, shared across scale experts to encode variable identity. We repeat E_{var} along the segment dimension to obtain $\widetilde{E}_{\text{var}} \in \mathbb{R}^{V \times P \times D}$, and project E_{scale} to dimension D to obtain $\widetilde{E}_{\text{scale}} \in \mathbb{R}^{V \times P \times D}$. We then concatenate the two representations and project them back to dimension D:

$$E_{\text{concat}} = \mathbf{f}(\widetilde{E}_{\text{scale}} \| \widetilde{E}_{\text{var}}) \in \mathbb{R}^{V \times P \times D}, \tag{8}$$

where $\|$ denotes the concatenation operation, and $\mathbf{f}$ is implemented as a single linear layer. Subsequently, we employ the Multi-Head Attention (MHA) mechanism (Vaswani et al., 2017) to capture the inter-variable dependencies, yielding the final representation at scale s:

$$E_s = \mathbf{MHA}(E_{\text{concat}}, E_{\text{concat}}, E_{\text{concat}}) \in \mathbb{R}^{V \times P \times D}. \tag{9}$$

Each scale expert uses an independent MHA to model scale-specific inter-variable dependencies.

4.2.3 EXPERT FUSION VIA GATED ROUTING

Given the scale-specific representations $\{E_s\}_{s=1}^{k}$, we use an MoE router conditioned on the soft-segment temporal representations C to adaptively combine them for each segment. We sum C along the temporal dimension to obtain $c \in \mathbb{R}^{V \times P \times D}$, which is then fed into the gating network. Following Chen et al. (2024), we introduce a noise term to enhance randomness:

$$G(c) = \text{Softmax}(cW_r + \epsilon \cdot \text{Softplus}(cW_{\text{noise}})), \quad \epsilon \sim \mathcal{N}(0, 1), \tag{10}$$

where W_r and $W_{\text{noise}} \in \mathbb{R}^{D \times k}$ are learnable weight matrices. The final multi-scale representation is computed as the weighted sum over all scale experts:

$$H = \sum_{s=1}^{k} G(c)_s E_s \in \mathbb{R}^{V \times P \times D}, \tag{11}$$

where $G(c)_s$ denotes the s-th element of the gating vector $G(c)$, representing the routing weight assigned to the s-th scale expert. Finally, we aggregate H along the cursor dimension P using a learned mapping $\psi(\cdot)$, implemented as an MLP, yielding $\bar{H} = \psi(H) \in \mathbb{R}^{V \times D}$.

To prevent expert collapse, where the MoE router tends to assign disproportionately high weights to a small subset of "popular" experts, we follow Liu et al. (2025b) and introduce a balance loss:

$$\mathcal{L}_{\text{imp}}(C) = W_{\text{imp}} \mathbf{CV}\left(\sum G(c)\right)^2, \tag{12}$$

where the summation is taken over all variables and segments, W_{imp} is a learnable weight parameter, and $\mathbf{CV}(\cdot)$ denotes the Coefficient of Variation.

4.3 PREDICTION HEAD

For each query $q_j = (t_j, v_j) \in \mathcal{Q}$, we concatenate the representation $\bar{H}_{v_j}$ of the target variable with the temporal embedding $\phi(t_j)$ of its query timestamp. An MLP decoder maps the concatenated representation to the predicted value (Shukla & Marlin, 2019):

$$\hat{z}_j = \text{MLP}\big([\bar{H}_{v_j} \parallel \phi(t_j)]\big), \qquad j = 1, \ldots, M_Q. \tag{13}$$

where $\|$ denotes concatenation. The resulting predictions are denoted by $\hat{\mathcal{Z}} = \{\hat{z}_j\}_{j=1}^{M_Q}$.

4.4 LOSS FUNCTIONS

We utilize the Mean Squared Error (MSE) as the primary objective function to minimize the discrepancy between the predicted values and the ground truth:

$$\mathcal{L}_{\text{mse}} = \frac{1}{M_Q} \sum_{j=1}^{M_Q} (z_j - \hat{z}_j)^2. \tag{14}$$

The total loss is defined as $\mathcal{L} = \mathcal{L}_{\text{mse}} + \gamma \mathcal{L}_{\text{dist}} + \beta \mathcal{L}_{\text{imp}}$. The coefficients γ and β control the relative contributions of the diversity loss $\mathcal{L}_{\text{dist}}$ and the expert balance loss $\mathcal{L}_{\text{imp}}$, respectively.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING

Datasets and Baselines. Following the methodology of Luo et al. (2025), we evaluate our model on four benchmark datasets: PhysioNet (Silva et al., 2012), MIMIC-III (Johnson et al., 2016), Human Activity, and USHCN (Menne et al., 2015). These datasets span diverse domains, including healthcare informatics, biomechanics, and climate science. For all datasets, we adhere to the preprocessing protocols described by Zhang et al. (2024), where samples are randomly partitioned into training, validation, and test sets using a 6:2:2 ratio. Detailed dataset statistics are provided in Appendix A. We employ Mean Squared Error (MSE) and Mean Absolute Error (MAE) to evaluate predictive performance, with formal definitions given in Appendix B.

To ensure a comprehensive and fair comparison, we benchmark our proposed method against 22 state-of-the-art (SOTA) baselines across various domains and tasks, categorized as follows: (1) Multivariate Time Series (MTS) Forecasting: iTransformer (Liu et al., 2024), ModernTCN (Luo & Wang, 2024), TimesNet (Wu et al., 2023), PatchTST (Nie et al., 2023), TimeFilter (Hu et al., 2025), (2) Multi-scale methods for MTS: Pathformer (Chen et al., 2024), TimeMixer (Wang et al., 2024), MSGNet (Cai et al., 2024), MICN (Wang et al., 2023), (3) IMTS Forecasting: tPatchGNN (Zhang et al., 2024), GraFiTi (Yalavarthi et al., 2024), CRU (Schirmer et al., 2022), mTAND (Shukla & Marlin, 2021), Neural Flows (Biloš et al., 2021), Latent ODEs (Rubanova et al., 2019), Raindrop (Zhang et al., 2021), GRU-D (Che et al., 2018), HyperIMTS (Li et al., 2025), APN (Liu et al., 2026), (4) Multi-scale methods for IMTS: Warpformer (Zhang et al., 2023), Hi-Patch (Luo et al., 2025), ReIMTS (Li et al., 2026). A detailed description of each baseline and its configuration is provided in Appendix C.

Implementation Details. We employ the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.001. Training is terminated early if the validation loss fails to decrease for 20 consecutive epochs. To ensure statistical robustness, all experiments are conducted using five different random seeds, with the mean and standard deviation reported. All models are implemented using the PyTorch library and trained on two NVIDIA GeForce RTX 3090 24GB GPUs. Detailed hyperparameter configurations can be found in Appendix D.

5.2 MAIN RESULTS

Table 1 reports the forecasting results under the default observation and forecast horizons, with additional settings provided in Appendices E.1 and E.2. Overall, SoftScale achieves state-of-the-art performance in most settings across the four benchmarks. Most regular time-series baselines show limited effectiveness because they are not designed to handle irregular observations. Existing multi-scale IMTS methods (Zhang et al., 2023; Luo et al., 2025; Li et al., 2026) apply a fixed scale configuration to all samples and periods, limiting their ability to adapt scales to varying temporal patterns. In contrast, SoftScale learns sample- and period-specific scale preferences and adaptively combines scale-specific representations for different periods. Beyond forecasting performance, we also evaluate the computational efficiency of SoftScale in Appendix E.3, showing that it maintains relatively low computational costs. Computational complexity and scalability are further analyzed in Appendix E.4.

5.3 ABLATION STUDY

To assess the contribution of each component, we conduct an ablation study using seven variants of **SoftScale**: (1) **w/o Multi-Expert** retains one single-scale expert; (2) **w/o Multi-Scale** uses three experts with the same scale; (3) **w/o MoE** replaces gated fusion with unweighted summation; (4) **w/o ACSS** removes the Adaptive Cursor-based Soft Segment (ACSS) module introduced in Section 4.1; (5) **w/o Cursor Drift** removes sample-dependent cursor offsets; (6) **w/o VA** removes inter-variable attention; and (7) **w/o IVA** shares MHA parameters across scale experts.

As shown in Table 2, every variant increases the mean error on all four datasets. The degradations without Multi-Expert or Multi-Scale validate the benefit of scale diversity, while those without ACSS or Cursor Drift support sample-adaptive soft segmentation and cursor adjustment. Removing MoE,

Table 1: Method benchmarking on IMTS forecasting. The best results are highlighted in **bold**, and the second-best results are underlined. The results are presented as (Mean ± Std). '−' indicates a numerical overflow error. The horizons are reported as observation → forecast.

	Methods	Human Activity 2000 ms → 2000 ms		USHCN 24 months → 12 months		PhysioNet 24 h → 24 h		MIMIC-III 24 h → 24 h	
		MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}
Regular	iTransformer	7.49 ± 4.72	6.08 ± 2.17	6.19 ± 0.12	4.01 ± 0.12	53.55 ± 19.59	16.87 ± 4.51	7.24 ± 0.50	21.24 ± 0.97
	ModernTCN	5.26 ± 0.06	4.96 ± 0.04	6.19 ± 0.07	3.75 ± 0.03	28.99 ± 11.06	5.94 ± 0.20	10.07 ± 9.79	10.76 ± 1.10
	TimesNet	5.38 ± 0.30	5.32 ± 0.20	5.84 ± 0.06	3.78 ± 0.10	9.30 ± 0.70	5.50 ± 0.34	2.34 ± 0.54	8.09 ± 0.10
	PatchTST	7.25 ± 0.29	6.26 ± 0.17	6.55 ± 0.12	4.20 ± 0.06	25.56 ± 4.42	10.90 ± 1.08	7.24 ± 0.65	19.80 ± 0.87
	TimeFilter	5.29 ± 0.06	4.92 ± 0.04	7.16 ± 0.31	4.34 ± 0.16	18.45 ± 1.12	7.09 ± 0.19	3.26 ± 0.90	10.43 ± 1.25
	Pathformer	4.67 ± 0.22	4.58 ± 0.17	–	–	6.75 ± 0.41	4.61 ± 0.20	–	–
	TimeMixer	5.39 ± 0.54	5.05 ± 0.38	5.93 ± 0.05	3.77 ± 0.03	13.98 ± 0.31	6.88 ± 0.09	4.78 ± 0.09	14.29 ± 0.06
	MSGNet	7.90 ± 0.36	6.85 ± 0.13	6.00 ± 0.03	3.85 ± 0.03	9.84 ± 0.29	5.79 ± 0.10	2.65 ± 0.18	9.10 ± 0.27
	MICN	7.57 ± 0.05	6.43 ± 0.02	5.99 ± 0.11	3.69 ± 0.08	10.34 ± 0.24	6.00 ± 0.14	2.36 ± 0.06	8.43 ± 0.22
Irregular	Warpformer	3.60 ± 0.08	3.81 ± 0.03	5.10 ± 0.07	3.13 ± 0.12	5.04 ± 0.14	3.72 ± 0.06	1.76 ± 0.30	7.27 ± 0.15
	Raindrop	5.57 ± 0.34	5.15 ± 0.11	7.61 ± 0.02	4.61 ± 0.05	10.63 ± 0.29	6.02 ± 0.19	2.31 ± 0.07	8.61 ± 0.12
	GRU-D	5.93 ± 0.10	5.66 ± 0.66	5.36 ± 0.12	3.25 ± 0.07	5.76 ± 0.34	4.53 ± 0.15	2.35 ± 0.06	8.34 ± 0.22
	tPatchGNN	3.71 ± 0.20	3.89 ± 0.16	6.23 ± 0.10	3.83 ± 0.60	5.06 ± 0.10	3.75 ± 0.07	1.97 ± 0.05	7.76 ± 0.22
	GraFiTi	4.59 ± 0.04	4.45 ± 0.04	5.01 ± 0.03	3.14 ± 0.06	5.11 ± 0.19	3.96 ± 0.09	1.75 ± 0.04	7.28 ± 0.13
	CRU	4.12 ± 0.08	4.43 ± 0.06	6.64 ± 0.95	4.08 ± 0.51	6.43 ± 0.62	4.51 ± 0.16	2.23 ± 0.03	7.99 ± 0.04
	mTAND	4.38 ± 0.37	4.59 ± 0.29	5.07 ± 0.03	3.09 ± 0.03	6.18 ± 0.31	4.44 ± 0.19	2.15 ± 0.05	8.00 ± 0.06
	NeuralFlow	5.47 ± 0.49	5.35 ± 0.28	5.48 ± 0.37	3.56 ± 0.37	7.68 ± 0.37	4.84 ± 0.19	2.34 ± 0.05	8.09 ± 0.09
	Latent-ODE	5.04 ± 0.46	5.11 ± 0.29	5.23 ± 0.04	3.35 ± 0.02	6.85 ± 0.28	4.77 ± 0.17	2.11 ± 0.15	7.76 ± 0.08
	Hi-Patch	3.29 ± 0.04	3.70 ± 0.04	5.02 ± 0.05	3.07 ± 0.08	4.86 ± 0.03	3.62 ± 0.07	1.75 ± 0.26	7.24 ± 0.18
	HyperIMTS	3.37 ± 0.07	3.69 ± 0.04	5.02 ± 0.04	3.12 ± 0.11	4.76 ± 0.15	3.65 ± 0.05	1.74 ± 0.16	6.92 ± 0.17
	APN	3.74 ± 0.03	3.85 ± 0.04	6.03 ± 0.80	3.88 ± 0.46	19.28 ± 26.95	7.55 ± 7.04	1.94 ± 0.05	7.55 ± 0.10
	ReIMTS	3.68 ± 0.03	3.76 ± 0.04	5.25 ± 0.10	3.33 ± 0.18	5.00 ± 0.04	3.84 ± 0.15	–	–
	SoftScale	**3.19 ± 0.02**	**3.63 ± 0.01**	**4.98 ± 0.07**	**3.00 ± 0.07**	**4.57 ± 0.07**	**3.51 ± 0.05**	**1.67 ± 0.03**	**6.85 ± 0.08**

Table 2: Ablation results on four datasets. The results are presented as Mean ± Std over five random seeds.

Methods	Human Activity		USHCN		PhysioNet		MIMIC-III	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}
Complete	**3.19 ± 0.02**	**3.63 ± 0.01**	**4.98 ± 0.07**	**3.00 ± 0.07**	**4.57 ± 0.07**	**3.51 ± 0.05**	**1.67 ± 0.03**	**6.85 ± 0.08**
w/o Multi-Expert	3.31 ± 0.11	3.72 ± 0.06	5.11 ± 0.06	3.11 ± 0.03	4.69 ± 0.10	3.53 ± 0.05	1.70 ± 0.05	6.96 ± 0.13
w/o Multi-Scale	3.23 ± 0.02	3.68 ± 0.02	5.06 ± 0.06	3.08 ± 0.07	4.62 ± 0.12	3.53 ± 0.06	1.71 ± 0.04	6.89 ± 0.09
w/o MoE	3.23 ± 0.03	3.68 ± 0.03	5.04 ± 0.06	3.04 ± 0.06	4.73 ± 0.16	3.53 ± 0.02	1.75 ± 0.03	6.97 ± 0.24
w/o ACSS	3.22 ± 0.04	3.67 ± 0.04	5.03 ± 0.05	3.13 ± 0.05	4.78 ± 0.07	3.59 ± 0.03	1.76 ± 0.05	7.12 ± 0.05
w/o Cursor Drift	3.23 ± 0.01	3.66 ± 0.02	5.02 ± 0.04	3.06 ± 0.07	4.75 ± 0.19	3.60 ± 0.11	1.70 ± 0.04	6.95 ± 0.04
w/o VA	3.53 ± 0.05	3.73 ± 0.01	5.11 ± 0.06	3.14 ± 0.12	4.81 ± 0.05	3.54 ± 0.03	1.80 ± 0.03	7.34 ± 0.17
w/o IVA	3.23 ± 0.02	3.67 ± 0.04	5.08 ± 0.11	3.04 ± 0.10	4.74 ± 0.05	3.54 ± 0.05	1.69 ± 0.03	6.90 ± 0.10

VA, or IVA further confirms the value of adaptive fusion and scale-specific inter-variable modeling. Hyperparameter analyses are provided in Appendices E.5–E.7.

5.4 Case study

To further examine the learned scale preferences, Figure 3 presents a case study with two samples in a setting with two time cursors and three scale experts. In Sample A, Soft-segment 1 focuses on the densely observed early period with stronger local fluctuations and favors finer scales, whereas Soft-segment 2 focuses on the sparser and smoother later period and favors coarser scales. This shift in scale preferences follows the change in temporal patterns across periods. In Sample B, the observations are more evenly spaced and exhibit relatively smoother temporal evolution, while the two soft segments retain similar and more balanced scale combinations. Together, these cases show that SoftScale matches scale combinations to sample- and period-specific temporal patterns, enabling adaptive multi-scale modeling for IMTS. Additional visualizations of the learned multi-scale views are provided in Appendix F.

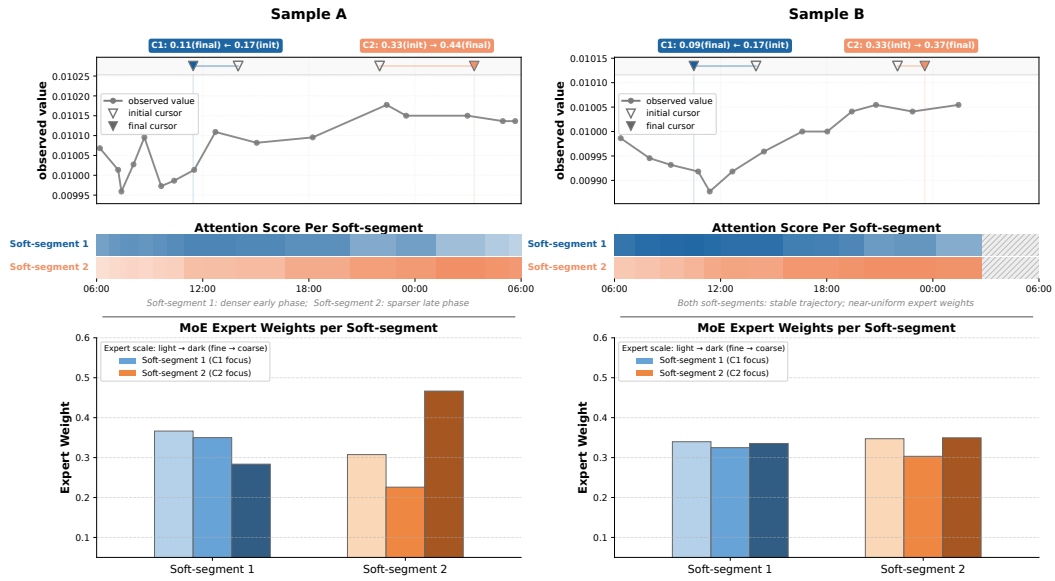


Figure 3: Case study of SoftScale on the pH variable from two patient samples in the PhysioNet dataset (Silva et al., 2012). **Top:** observed values and the initial and final positions of two data-dependent time cursors. **Middle:** continuous assignment weights of observations to the two soft segments. **Bottom:** MoE routing weights over the fine, intermediate, and coarse experts for each soft segment.

Routing Validation. To complement Figure 3, we perturb only the routing weights on PhysioNet during inference. Sample Shuffle exchanges routing weights across samples, while Expert Shuffle permutes them across scale experts in one or both soft segments. All perturbations increase errors, supporting sample-specific routing and period-specific scale matching, with the largest degradation when both segments are modified.

Table 3: Inference-time routing perturbations on PhysioNet.

Setting	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$
Sample Shuffle	4.72 ± 0.11	3.56 ± 0.07
Both Segments	4.77 ± 0.12	3.60 ± 0.09
Segment 1	4.62 ± 0.10	3.53 ± 0.07
Segment 2	4.64 ± 0.10	3.54 ± 0.07
Original	$\mathbf{4.57 \pm 0.07}$	$\mathbf{3.51 \pm 0.05}$

6 Conclusion and Limitations

In this paper, we proposed SoftScale, an adaptive soft-segmented multi-scale framework for IMTS forecasting. SoftScale uses data-dependent time cursors to construct sample-adaptive soft segments from raw IMTS. A multi-scale experts network then captures temporal and inter-variable dependencies at multiple scales, while an MoE router learns sample- and period-specific scale preferences. Experimental results demonstrate that SoftScale outperforms existing methods in most settings while maintaining competitive computational efficiency.

Despite its effectiveness, SoftScale has several limitations. Although SoftScale adaptively learns scale preferences and cursor positions, its scale pool and number of time cursors remain predefined. Moreover, it is not yet a plug-and-play module. Future work will explore data-driven scale construction, adaptive cursor allocation, and a general-purpose extension for IMTS modeling.

AI Use Statement

In this work, we used generative AI tools to assist with translation and the interpretation of results. We did not use generative AI tools for any other tasks requiring disclosure, such as developing theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, or implementing methods; tasks not relevant to this work are not applicable. Beyond these disclosed uses, generative AI tools were used only for language polishing to improve clarity,

concision, and grammar. All AI-assisted translations, result interpretations, and language edits were reviewed and verified by the authors against the underlying manuscript and reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems*, 34:21325–21337, 2021.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11141–11149, 2024.

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.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *ICLR 2024: The Twelfth International Conference on Learning Representations.*, 2024.

Zipeng Chen, Qianli Ma, and Zhenxi Lin. Time-aware multi-scale rnns for time series modeling. In *IJCAI*, pp. 2285–2291, 2021.

Mingyue Cheng, Qi Liu, Zhiding Liu, Zhi Li, Yucong Luo, and Enhong Chen. Formertime: Hierarchical multi-scale representations for multivariate time series classification. In *Proceedings of the ACM web conference 2023*, pp. 1437–1445, 2023.

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

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pp. 1651–1661. PMLR, 2020.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=490VcNtjh7`.

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

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Xiang Lan, Hanshu Yan, Shenda Hong, and Mengling Feng. Towards enhancing time series contrastive learning: A dynamic bad pair mining approach. In *The Twelfth International Conference on Learning Representations*, 2024.

Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. Hyperimts: Hypergraph neural network for irregular multivariate time series forecasting. In *International Conference on Machine Learning*, pp. 35502–35518. PMLR, 2025.

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.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *International Conference on Machine Learning*, pp. 37797–37814. PMLR, 2025.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2022.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, et al. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In *International Conference on Machine Learning*, pp. 38940–38962. PMLR, 2025a.

Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23873–23881, 2026.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhen Liu, Yicheng Luo, Boyuan Li, Emadeldeen Eldele, Min Wu, and Qianli Ma. Learning soft sparse shapes for efficient time-series classification. In *International Conference on Machine Learning*, pp. 39032–39059. PMLR, 2025b.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.

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

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-patch: Hierarchical patch gnn for irregular multivariate time series. In *Forty-second International Conference on Machine Learning*, 2025.

Qianli Ma, Zhenxi Lin, Enhuan Chen, and Garrison Cottrell. Temporal pyramid recurrent neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5061–5068, 2020.

MJ Menne, CN Williams Jr, RS Vose, and Data Files. Long-term daily climate records from stations across the contiguous united states, 2015.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

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

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

Mohammad Amin Shabani, Amir H Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, volume 2025, pp. 34635–34667, 2025.

Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

Satya Narayan Shukla and Benjamin Marlin. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *International Conference on Learning Representations*, 2022.

Satya Narayan Shukla and Benjamin M Marlin. Interpolation-prediction networks for irregularly sampled time series. *arXiv preprint arXiv:1909.07782*, 2019.

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

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*, pp. 245–248. IEEE, 2012.

Qingxiong Tan, Mang Ye, Grace Lai-Hung Wong, and Pong Chi Yuen. Cooperative joint attentive network for patient outcome prediction on irregular multi-rate multivariate health data. In *IJCAI*, pp. 1586–1592, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. In *International Conference on Learning Representations*, pp. 1–23. Curran Associates Inc., 2024.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Yinjun Wu, Jingchao Ni, Wei Cheng, Bo Zong, Dongjin Song, Zhengzhang Chen, Yanchi Liu, Xuchao Zhang, Haifeng Chen, and Susan B Davidson. Dynamic gaussian mixture based deep generative model for robust forecasting on sparse multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 651–659, 2021.

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.

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

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.

Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Spatio-temporal graph structure learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1177–1185, 2020.

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

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2021.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A FURTHER DETAILS OF DATASETS

We use 4 irregularly sampled multivariate time series datasets to evaluate the performance of our model and baseline models. The dataset statistics are summarized in Table 4.

Table 4: Dataset statistics for forecasting.

Datasets	# Samples	# Variables	Avg Sample Length	Missing Ratio
Human Activity	5,400	12	120	75.0%
USHCN	26,736	5	163	77.9%
PhysioNet	11,988	41	74	88.4%
MIMIC-III	23,457	96	46	96.7%

We evaluate on four benchmark datasets following the preprocessing protocols of Zhang et al. (2024). Each dataset is split into training, validation, and test sets at a 6:2:2 ratio. We adopt three historical/forecasting horizon configurations for each dataset.

PhysioNet (Silva et al., 2012) The PhysioNet dataset comprises records of 12,000 ICU patients. Each record contains up to 41 clinical variables measured irregularly over the first 48 hours following ICU admission. After preprocessing, 11,988 samples with 41 variables are retained. For the forecasting task, we use the first 24/36/12 hours of each patient's record as the observation window to predict the clinical values in the subsequent 24/12/36 hours.

MIMIC-III (Johnson et al., 2016) MIMIC-III is a large-scale clinical database containing irregularly sampled multivariate time series from 23,457 ICU patients. Each patient's record spans the first 48 hours of ICU admission and encompasses 96 clinical variables. Following the same protocol, we use the first 24/36/12 hours as the observed window to forecast the queried values in the subsequent 24/12/36 hours.

Human Activity The Human Activity dataset captures the motion of five subjects performing a variety of daily activities, recorded by inertial sensors placed on the ankles, belts, and chests of the participants. The raw recordings consist of 12 three-dimensional positional variables sampled at irregular intervals. To construct a realistic forecasting benchmark, the continuous recordings are chunked into 5,400 non-overlapping IMTS instances, each spanning 4,000 milliseconds. The first 3,000/2,000/1,000 milliseconds of each instance serve as the observation window, and the model is tasked with predicting sensor positions over the remaining 1,000/2,000/3,000 milliseconds.

USHCN (Menne et al., 2015) USHCN dataset provides long-term daily climate observations from meteorological stations distributed across the contiguous United States, covering five key climate variables. Following established preprocessing procedures, we focus on records from 1,114 stations spanning the period from 1996 to 2000, yielding 26,736 IMTS instances. For forecasting, each instance uses the preceding 24 months of climate data as the observation window to predict climate conditions over the next 1/6/12 months.

B PERFORMANCE METRICS

MSE MSE (Mean Squared Error) measures the average of the squared differences between predicted and actual values. The calculation formula is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{15}$$

where y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n is the number of observations. A smaller MSE indicates better model performance. Since errors are squared, MSE is sensitive to large errors or outliers.

MAE MAE (Mean Absolute Error) measures the average of the absolute differences between predicted and actual values. The calculation formula is:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{16}$$

where y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n is the number of observations. A smaller MAE indicates better model performance. Compared to MSE, MAE is less sensitive to outliers and provides a straightforward average measure of error.

C BASELINES

For methods designed for regular MTS, we organize each sample into a $V \times T_h$ history matrix and a $V \times T_f$ forecasting matrix, where V represents the maximum number of variables and T_h/T_f represents the historical/forecasting length of the sample. Unobserved positions are filled with zeros, the sequence length is set to the maximum historical length across all samples, and the forecasting length is set to the maximum forecasting length across all samples.

iTransformer (Liu et al., 2024) uses inverted Transformers, applying the attention mechanism across the variable dimension rather than the time dimension, for effective multivariate time series forecasting. We use the following setting in our experiment: `e_layers = 3`, `d_layers = 1`, `factor = 3`, `d_model = 512`, `d_ff = 512`.

ModernTCN (Luo & Wang, 2024) is a modern pure convolution structure that revisits and modernizes traditional TCN designs for general time series analysis. We use the implementation at `https://github.com/luodhhh/ModernTCN` and the following setting in our experiment: `ffn_ratio = 8`, `patch_size = 8`, `patch_stride = 4`, `num_blocks = 1`, `large_size = 51`, `small_size = 5`, `dims = 64`.

TimesNet (Wu et al., 2023) transforms 1D time series into 2D space to model temporal variations through a 2D-variation framework for general time series analysis. We use the following setting in our experiment: `e_layers = 2`, `d_layers = 1`, `factor = 3`, `d_model = 256`, `d_ff = 512`, `top_k = 5`.

PatchTST (Nie et al., 2023) is a Transformer-based model that segments time series into subseries-level patches and applies channel independence for long-term forecasting. We use the following setting in our experiment: `e_layers = 2`, `d_layers = 1`, `factor = 3`, `patch_len = 16`, `stride = 8`.

TimeFilter (Hu et al., 2025) is a patch-specific spatial-temporal graph filtration method for time series forecasting. We use the following setting in our experiment: `e_layers = 2`, `d_layers = 1`, `d_model = 512`, `d_ff = 2048`.

Pathformer (Chen et al., 2024) is a multi-scale Transformer architecture with adaptive pathways that simultaneously models temporal dependencies at multiple scales. We use the implementation at `https://github.com/decisionintelligence/pathformer` and the following setting in our experiment: `k = 2`, `layer_nums = 3`, `d_model = 16`, `d_ff = 64`.

TimeMixer (Wang et al., 2024) achieves complementary predictive capabilities by decomposing and mixing multi-scale temporal variations in both the past and future. We use the following setting in our experiment: `e_layers = 3`, `d_layers = 1`, `factor = 3`, `d_model = 16`, `d_ff = 32`, `down_sampling_layers = 3`, `down_sampling_window = 2`.

MSGNet (Cai et al., 2024) learns multi-scale inter-series correlations for multivariate time series forecasting via a combination of spectral graph convolutions and adaptive graph structures. We use the implementation at `https://github.com/YoZhibo/MSGNet` and the following setting in our experiment: `e_layers = 2`, `d_model = 512`, `d_ff = 64`, `n_heads = 8`, `top_k = 5`, `dropout = 0.1`, `nums_kernels = 6`, `conv_channel = 32`, `skip_channel = 32`, `gcn_depth = 2`, `propalpha = 0.3`, `node_dim = 10`, `gcn_dropout = 0.3`.

MICN (Wang et al., 2023) achieves multi-scale local and global context modeling for long-term time series forecasting through a multi-scale isometric convolution architecture. We use the following setting in our experiment: `e_layers = 2`, `d_layers = 1`, `factor = 3`,

`d_model = 256`, `d_ff = 512`, `top_k = 5`, `decomp_kernel = 32`, `conv_kernel = 24`, `isometric_kernel = [18, 6]`.

tPatchGNN (Zhang et al., 2024) is a transformable patching graph neural network for irregular multivariate time series forecasting, which groups temporally local observations into patches and models inter-variable dependencies via graph neural networks.

GraFiTi (Yalavarthi et al., 2024) is a graph-based method for forecasting irregularly sampled time series, which constructs dynamic graphs to capture both temporal and inter-variable relationships.

CRU (Schirmer et al., 2022) models irregular time series with continuous recurrent units, using a Kalman filtering framework to handle non-uniform sampling in a principled probabilistic manner.

mTAND (Shukla & Marlin, 2021) is a multi-time attention network for irregularly sampled time series. It learns a continuous-time embedding and employs attention to produce fixed-length representations from sequences with a variable number of observations.

Neural Flows (Biloš et al., 2021) offers an efficient alternative to Neural ODEs by using normalizing flows to model continuous-time dynamics for irregularly sampled time series.

Latent ODEs (Rubanova et al., 2019) employs neural ordinary differential equations to model the latent state dynamics of irregularly sampled time series in continuous time.

Raindrop (Zhang et al., 2021) is a graph-guided neural network for irregularly sampled multivariate time series that simultaneously learns inter-sensor dynamics and temporal observation patterns.

GRU-D (Che et al., 2018) extends the Gated Recurrent Unit (GRU) to handle missing values and irregular sampling by incorporating masking and time-interval information directly into the recurrent update equations.

HyperIMTS (Li et al., 2025) is a hypergraph neural network for irregular multivariate time series forecasting, which uses hyperedges to capture higher-order inter-variable relationships beyond pairwise interactions.

APN (Liu et al., 2026) is a simple yet effective baseline for irregular time series forecasting that rethinks the modeling of IMTS through adaptive prediction networks.

Warpformer (Zhang et al., 2023) is the first multi-scale modeling approach specifically designed for irregular clinical time series. It employs a warping module to unify irregular observations at multiple predefined scales, followed by a Transformer encoder to capture temporal dependencies.

Hi-Patch (Luo et al., 2025) is a hierarchical patch graph neural network for irregular multivariate time series forecasting. It partitions observations into multi-scale patches in a hierarchical manner and models inter-variable dependencies via graph neural networks at each scale.

ReIMTS (Li et al., 2026) learns recursive multi-scale representations for irregular multivariate time series forecasting through a hierarchical recursive framework that progressively abstracts temporal information at increasing scales.

D HYPERPARAMETERS SETTINGS FOR SOFTSCALE

We search all hyperparameters in the grid for our proposed model SoftScale. Specifically, our model has a total of 4 hyperparameters: the number of soft segments P, the scale pool $\mathcal{R}$, the temporal encoding dimension D, and the number of MHA layers n_{layer}. We search P over the range $\{1, 2, 3, 4, 5\}$, the scale pool $\mathcal{R}$ over subsets of $\{128, 64, 32, 16, 8, 4, 2\}$, D in $\{16, 32, 64, 128\}$, and n_{layer} in $\{1, 2, 3\}$. For the loss function coefficients defined in Section 4.4, we fix $\gamma = 1\text{e}{-5}$ for the diversity loss $\mathcal{L}_{dist}$ and $\beta = 1\text{e}{-2}$ for the expert balance loss $\mathcal{L}_{imp}$ across all datasets. The best hyperparameters for each dataset are reported in the code.

E ADDITIONAL EXPERIMENTAL RESULTS

E.1 VARYING OBSERVATION HORIZONS AND FIXED FORECAST HORIZON

We conduct additional experiments to evaluate the model's ability to capture more temporal dependencies from long-term data on 4 datasets. Specifically:

- **Human Activity**: Fixed forecast length at 2000 ms, varied observation horizons as $\{500, 1000, 2000\}$ ms.
- **USHCN**: Fixed forecast length at 12 month, varied observation horizons as $\{6, 12, 24\}$ months.
- **PhysioNet**: Fixed forecast length at 24 hours, varied observation horizons as $\{6, 12, 24\}$ hours.
- **MIMIC-III**: Fixed forecast length at 24 hours, varied observation horizons as $\{6, 12, 24\}$ hours.

Figure 4 shows the results of our model and the three most competitive baseline methods: Warpformer, HyperIMTS, and Hi-Patch. It is evident that our method consistently improves in MAE as the historical length increases across all four datasets, demonstrating the model's capability to learn more temporal dependencies from longer-term data. Conversely, some existing IMTS methods do not have this ability, such as Warpformer on the USHCN and MIMIC-III datasets, where its MAE increases as the observation horizon extends from 6 months to 12 months and from 6 hours to 12 hours, respectively. These supplementary experiments further highlight the superiority of our SoftScale approach compared to prior methods.

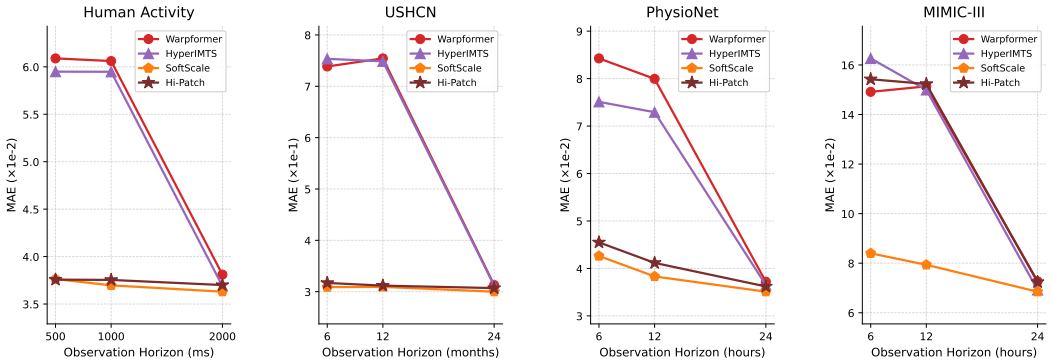


Figure 4: Forecasting performance with varying observation horizons and fixed forecast horizon across four datasets.

E.2 VARYING OBSERVATION AND FORECAST HORIZONS

Table 5: Forecasting results on Human Activity dataset under additional horizon settings. The best results are highlighted in **bold**, and the second-best results are underlined. Results are in the form of (Mean ± Std).

	Horizon	1000ms → 3000ms		3000ms → 1000ms	
	Metric	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}
Regular	iTransformer	5.58 ± 0.04	5.13 ± 0.05	3.97 ± 0.10	4.30 ± 0.08
	ModernTCN	12.12 ± 1.01	5.72 ± 0.07	3.99 ± 0.05	4.32 ± 0.04
	TimesNet	9.90 ± 0.42	7.34 ± 1.81	3.79 ± 0.05	4.28 ± 0.04
	PatchTST	8.97 ± 1.96	6.94 ± 0.94	5.21 ± 0.33	5.10 ± 0.20
	TimeFilter	6.72 ± 0.09	5.39 ± 0.08	4.73 ± 0.56	4.80 ± 0.34
	Pathformer	5.49 ± 0.12	5.05 ± 0.06	3.40 ± 0.16	3.65 ± 0.08
	TimeMixer	5.96 ± 0.19	5.36 ± 0.10	4.97 ± 0.31	5.02 ± 0.16
	MSGNet	8.29 ± 0.22	7.02 ± 0.12	6.32 ± 0.16	6.06 ± 0.09
	MICN	8.16 ± 0.12	6.78 ± 0.05	6.93 ± 0.12	6.04 ± 0.06
Irregular	Warpformer	4.26 ± 0.11	4.26 ± 0.04	2.61 ± 0.02	3.12 ± 0.01
	Raindrop	5.75 ± 0.33	5.37 ± 0.22	4.42 ± 0.25	4.65 ± 0.14
	GRU-D	6.14 ± 0.76	5.75 ± 0.49	3.94 ± 0.29	4.37 ± 0.21
	tPatchGNN	4.56 ± 0.08	4.32 ± 0.06	2.79 ± 0.09	3.24 ± 0.06
	GraFITi	4.91 ± 0.07	4.62 ± 0.03	3.03 ± 0.14	3.45 ± 0.10
	CRU	4.85 ± 0.09	4.86 ± 0.07	3.03 ± 0.04	3.60 ± 0.04
	mTAND	5.29 ± 0.32	5.12 ± 0.23	3.14 ± 0.09	3.71 ± 0.06
	NeuralFlow	6.01 ± 0.91	5.66 ± 0.60	4.29 ± 0.63	4.61 ± 0.43
	Latent-ODE	5.48 ± 0.21	5.33 ± 0.14	3.32 ± 0.10	3.91 ± 0.08
	Hi-Patch	4.21 ± 0.08	4.25 ± 0.07	2.57 ± 0.02	3.11 ± 0.03
	HyperIMTS	4.25 ± 0.12	4.30 ± 0.07	2.54 ± 0.04	$\mathbf{3.07 \pm 0.04}$
	APN	4.65 ± 0.04	4.35 ± 0.08	2.84 ± 0.01	3.21 ± 0.03
	ReIMTS	4.80 ± 0.16	4.45 ± 0.13	2.78 ± 0.02	3.14 ± 0.03
	SoftScale	$\mathbf{4.10 \pm 0.07}$	$\mathbf{4.21 \pm 0.05}$	$\mathbf{2.53 \pm 0.02}$	3.11 ± 0.03

Table 6: Performance of varying observation and forecast horizons on MIMIC-III dataset. The best results are highlighted in **bold**, and the second-best results are underlined. Results are in the form of (Mean ± Std).

	Horizon	36h → 12h		12h → 36h	
	Metric	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}
Regular	iTransformer	7.27 ± 0.04	21.52 ± 0.09	7.44 ± 0.21	21.47 ± 0.45
	ModernTCN	5.12 ± 0.59	11.41 ± 0.30	3.96 ± 0.23	10.79 ± 0.39
	TimesNet	2.02 ± 0.09	8.18 ± 0.22	2.39 ± 0.06	8.35 ± 0.20
	PatchTST	8.81 ± 0.96	22.97 ± 1.32	7.11 ± 0.65	19.96 ± 1.24
	TimeFilter	–	–	3.36 ± 0.63	10.75 ± 1.75
	Pathformer	–	–	2.45 ± 0.02	8.65 ± 0.09
	TimeMixer	4.31 ± 0.29	13.44 ± 0.28	6.31 ± 0.04	18.41 ± 0.06
	MSGNet	2.48 ± 0.12	9.42 ± 0.32	2.57 ± 0.03	9.02 ± 0.06
	MICN	2.15 ± 0.13	8.66 ± 0.26	2.39 ± 0.06	8.58 ± 0.11
Irregular	Warpformer	1.45 ± 0.10	6.74 ± 0.08	2.32 ± 0.04	8.14 ± 0.07
	Raindrop	2.21 ± 0.37	9.17 ± 0.49	2.36 ± 0.03	8.63 ± 0.11
	GRU-D	2.03 ± 0.13	8.14 ± 0.26	2.39 ± 0.02	8.43 ± 0.13
	tPatchGNN	1.44 ± 0.08	6.78 ± 0.14	2.35 ± 0.03	8.23 ± 0.08
	GraFITi	1.61 ± 0.27	7.16 ± 0.36	2.22 ± 0.05	8.13 ± 0.13
	CRU	2.00 ± 0.13	8.16 ± 0.26	2.34 ± 0.05	8.32 ± 0.13
	mTAND	2.01 ± 0.09	8.13 ± 0.23	2.29 ± 0.03	8.38 ± 0.08
	NeuralFlow	1.97 ± 0.12	8.39 ± 0.25	2.26 ± 0.08	8.29 ± 0.10
	Latent-ODE	1.90 ± 0.03	7.92 ± 0.17	2.38 ± 0.05	8.35 ± 0.13
	Hi-Patch	1.56 ± 0.10	6.71 ± 0.16	2.32 ± 0.02	8.11 ± 0.08
	HyperIMTS	1.39 ± 0.07	6.61 ± 0.18	$\mathbf{2.07 \pm 0.06}$	$\mathbf{7.47 \pm 0.17}$
	APN	$\mathbf{1.31 \pm 0.01}$	6.69 ± 0.09	2.35 ± 0.03	8.27 ± 0.20
	ReIMTS	–	–	–	–
	SoftScale	1.42 ± 0.08	$\mathbf{6.42 \pm 0.09}$	2.28 ± 0.02	7.88 ± 0.08

Table 7: Performance of varying observation and forecast horizons on USHCN dataset. The best results are highlighted in **bold**, and the second-best results are underlined. Results are in the form of (Mean ± Std). '−' indicates a numerical overflow error.

	Horizon	24months → 6months		24months → 1month	
	Metric	MSE×10^{-1}	MAE×10^{-1}	MSE×10^{-1}	MAE×10^{-1}
Regular	iTransformer	6.05 ± 0.03	3.89 ± 0.14	6.17 ± 0.07	4.18 ± 0.55
	ModernTCN	6.03 ± 0.10	3.68 ± 0.06	5.83 ± 0.13	3.58 ± 0.09
	TimesNet	5.68 ± 0.05	3.66 ± 0.06	5.62 ± 0.12	3.56 ± 0.12
	PatchTST	6.12 ± 0.03	4.01 ± 0.08	5.88 ± 0.10	3.66 ± 0.13
	TimeFilter	6.62 ± 0.12	4.07 ± 0.05	15.79 ± 5.33	6.87 ± 1.28
	TimeMixer	5.89 ± 0.07	3.65 ± 0.05	5.88 ± 0.10	3.59 ± 0.07
	MSGNet	5.76 ± 0.02	3.74 ± 0.08	5.75 ± 0.06	3.64 ± 0.09
	MICN	6.02 ± 0.03	3.85 ± 0.04	5.99 ± 0.11	3.69 ± 0.08
Irregular	Warpformer	5.12 ± 0.03	3.13 ± 0.08	5.09 ± 0.03	3.10 ± 0.04
	Raindrop	7.01 ± 0.49	4.24 ± 0.33	5.64 ± 0.10	3.29 ± 0.03
	GRU-D	5.29 ± 0.09	3.34 ± 0.09	5.17 ± 0.06	3.21 ± 0.05
	tPatchGNN	5.23 ± 0.02	3.24 ± 0.19	5.00 ± 0.03	3.07 ± 0.05
	GraFITi	5.12 ± 0.14	3.09 ± 0.10	5.07 ± 0.03	2.97 ± 0.04
	CRU	6.77 ± 1.04	4.11 ± 0.61	5.15 ± 0.50	3.18 ± 0.03
	mTAND	5.16 ± 0.10	3.10 ± 0.07	5.03 ± 0.05	3.00 ± 0.06
	NeuralFlow	5.52 ± 0.05	3.46 ± 0.05	5.41 ± 0.05	3.35 ± 0.06
	Latent-ODE	5.18 ± 0.04	3.36 ± 0.04	5.16 ± 0.04	3.21 ± 0.07
	Hi-Patch	5.07 ± 0.17	3.06 ± 0.06	$\mathbf{4.94 \pm 0.05}$	$\mathbf{2.96 \pm 0.04}$
	HyperIMTS	5.30 ± 0.15	3.23 ± 0.11	5.12 ± 0.19	3.10 ± 0.11
	APN	5.82 ± 0.21	3.58 ± 0.09	5.52 ± 0.03	3.43 ± 0.16
	ReIMTS	5.53 ± 0.11	3.42 ± 0.17	5.16 ± 0.05	3.31 ± 0.21
	SoftScale	$\mathbf{5.03 \pm 0.05}$	$\mathbf{2.99 \pm 0.08}$	5.10 ± 0.09	3.09 ± 0.10

Table 8: Performance of varying observation and forecast horizons on PhysioNet dataset. The best results are highlighted in **bold**, and the second-best results are underlined. Results are in the form of (Mean ± Std).

	Horizon	36h → 12h		12h → 36h	
	Metric	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}
Regular	iTransformer	56.83 ± 21.17	17.05 ± 5.21	54.17 ± 17.91	17.03 ± 4.18
	ModernTCN	24.87 ± 9.44	6.58 ± 0.58	31.48 ± 3.96	6.73 ± 0.53
	TimesNet	9.43 ± 0.53	5.55 ± 0.20	9.26 ± 0.18	5.58 ± 0.10
	PatchTST	26.13 ± 1.75	11.25 ± 0.29	25.63 ± 1.51	10.70 ± 0.38
	TimeFilter	39.10 ± 10.50	10.36 ± 1.39	24.08 ± 3.58	8.15 ± 0.33
	Pathformer	6.85 ± 0.42	4.61 ± 0.18	8.19 ± 0.28	5.03 ± 0.07
	TimeMixer	12.52 ± 0.45	6.47 ± 0.21	19.86 ± 0.17	8.38 ± 0.18
	MSGNet	10.44 ± 0.44	5.94 ± 0.08	9.91 ± 0.12	5.74 ± 0.08
	MICN	10.98 ± 0.31	6.01 ± 0.11	10.24 ± 0.14	5.85 ± 0.06
Irregular	Warpformer	4.17 ± 0.13	3.38 ± 0.08	6.51 ± 0.12	4.24 ± 0.04
	Raindrop	10.67 ± 0.33	5.87 ± 0.20	10.24 ± 0.18	5.83 ± 0.10
	GRU-D	6.85 ± 0.37	4.88 ± 0.18	7.80 ± 0.22	5.13 ± 0.13
	tPatchGNN	4.22 ± 0.09	3.38 ± 0.04	6.45 ± 0.11	4.24 ± 0.09
	GraFITi	4.58 ± 0.11	3.65 ± 0.05	6.30 ± 0.14	4.38 ± 0.12
	CRU	6.74 ± 0.21	4.82 ± 0.11	7.66 ± 0.14	4.97 ± 0.05
	mTAND	5.61 ± 0.31	4.15 ± 0.09	7.46 ± 0.19	4.85 ± 0.05
	NeuralFlow	8.87 ± 1.00	5.43 ± 0.18	7.98 ± 0.57	5.08 ± 0.24
	Latent-ODE	6.99 ± 0.24	4.74 ± 0.11	7.28 ± 0.13	4.83 ± 0.07
	Hi-Patch	4.16 ± 0.08	3.31 ± 0.06	6.30 ± 0.06	4.12 ± 0.05
	HyperIMTS	3.95 ± 0.09	3.37 ± 0.08	$\mathbf{5.67 \pm 0.08}$	4.10 ± 0.07
	APN	18.12 ± 27.72	7.20 ± 7.15	10.30 ± 4.91	5.73 ± 1.79
	ReIMTS	4.55 ± 0.55	3.65 ± 0.39	6.59 ± 0.42	4.45 ± 0.41
	SoftScale	$\mathbf{3.76 \pm 0.14}$	$\mathbf{3.09 \pm 0.06}$	6.07 ± 0.10	$\mathbf{3.97 \pm 0.04}$

E.3 Efficiency analysis

We compared 6 leading IMTS models: Warpformer (Zhang et al., 2023), GraFITi (Yalavarthi et al., 2024), Hi-Patch (Luo et al., 2025), HyperIMTS (Li et al., 2025), ReIMTS (Li et al., 2026), and Latent-ODE (Rubanova et al., 2019). All models are evaluated using the same batch size (16 for Human Activity, and 32 for PhysioNet, MIMIC-III, and USHCN) to assess their training time per epoch and MAE. The results shown in Figure 5 indicate that our method achieves competitive or superior MAE across all four datasets, while its training time consistently ranks among the fastest, placing 1st or 2nd on average. Notably, SoftScale models asynchronous correlations and multi-scale features overlooked by prior methods, while incurring no significant computational overhead. Our model's training time remains in the same order of magnitude as the fastest baselines, whereas methods such as Latent-ODE and ReIMTS require one to two orders of magnitude more time per epoch. In addition, on the MIMIC-III dataset, both Hi-Patch and ReIMTS run out of memory, while our model operates efficiently with competitive accuracy. In this context, we believe that our model achieves a favorable balance between predictive accuracy and computational efficiency, making it practical for real-world IMTS applications.

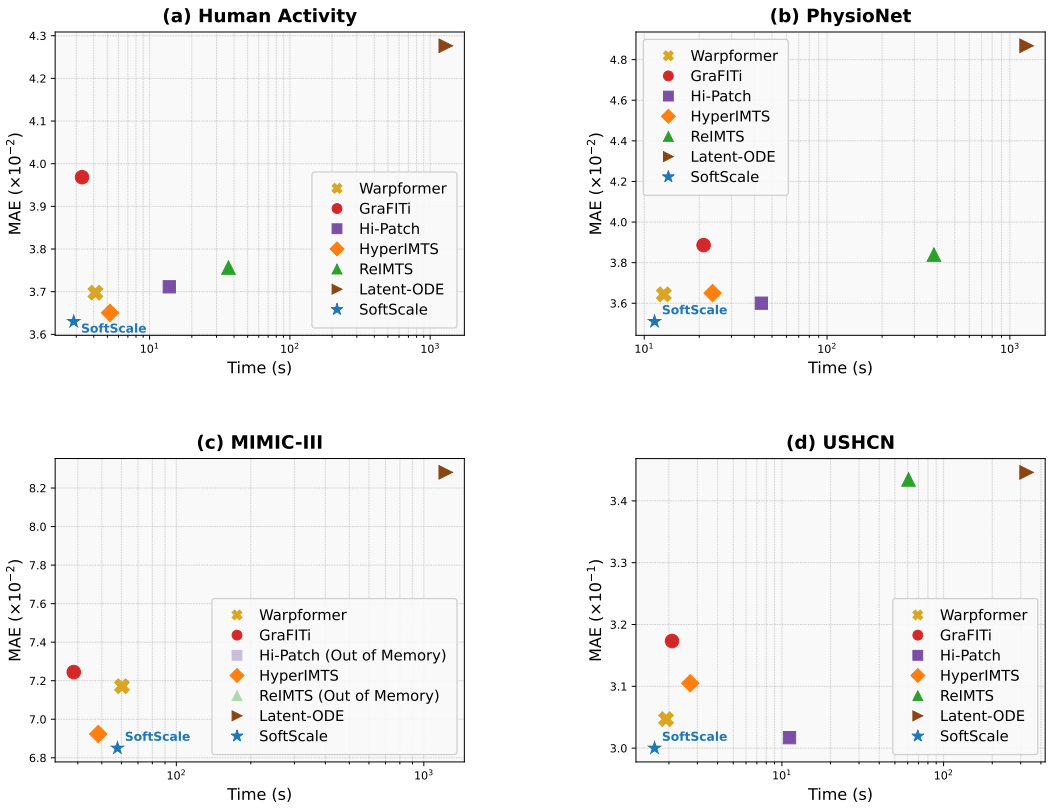


Figure 5: Comparison of IMTS models in terms of efficiency: training time per epoch against error metric.

E.4 Computational complexity and scalability

Let T, V, P, k, and D denote the numbers of observations, variables, time cursors, scale experts, and embedding dimensions, respectively, and let r_s denote the sequence length at the s-th scale. The dominant computation comes from cursor-to-timestamp attention, scale-specific temporal attention, and inter-variable attention:

$$\mathcal{O}\left(VPTD + \sum_{s=1}^{k}\left(VPTr_sD + PV^2D\right)\right).$$

Table 9: Empirical scalability of SoftScale.

Varied factor	Tested range	Inference time (ms)	Total peak memory (MB)	α: Time / Peak / Activation
Observations	$24 \rightarrow 384$ (16×)	$7.50 \pm 1.22 \rightarrow 32.22 \pm 1.99$ (4.30×)	$92.1 \rightarrow 903.5$ (9.81×)	0.55/0.83/0.99
Variables	$16 \rightarrow 256$ (16×)	$7.45 \pm 1.72 \rightarrow 36.46 \pm 2.54$ (4.89×)	$89.2 \rightarrow 914.8$ (10.25×)	0.64/0.84/1.00
Time cursors	$1 \rightarrow 16$ (16×)	$6.22 \pm 0.11 \rightarrow 34.35 \pm 1.99$ (5.52×)	$91.3 \rightarrow 906.1$ (9.92×)	0.55/0.83/1.00
Scale experts	$1 \rightarrow 8$ (8×)	$3.65 \pm 0.62 \rightarrow 17.99 \pm 1.94$ (4.93×)	$250.73 \rightarrow 259.14$ (1.03×)	0.77/0.02/0.00

For fixed D and sequence lengths, the computation grows linearly with T and P, approximately linearly with k, and includes a quadratic term in V due to inter-variable attention.

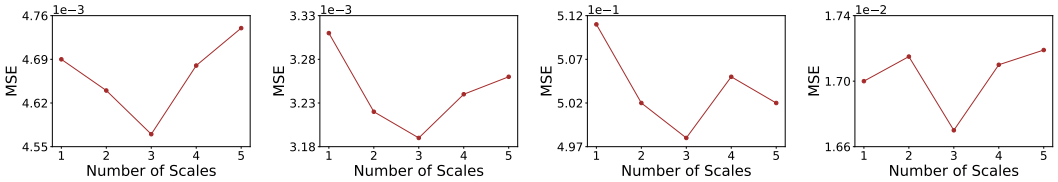
We further evaluate practical scalability by varying T, V, P, and k separately while holding the other factors fixed. For each measured quantity y and varied factor x, we fit $y = cx^{\alpha}$ in log-log space. Here, $\alpha \approx 1$, $0 < \alpha < 1$, and $\alpha \approx 0$ indicate linear, sublinear, and approximately constant growth, respectively. Incremental peak activation memory is defined as the peak allocated memory during inference minus the resident memory before the forward pass.

The three scaling exponents in Table 9 correspond to inference time, total peak memory, and incremental peak activation memory, respectively. For observations, variables, and time cursors, the fitted exponents indicate sublinear growth in inference time (0.55–0.64) and total peak memory (0.83–0.84), but approximately linear growth in incremental peak activation memory (0.99–1.00) within the tested ranges. The fitted exponents describe practical behavior under the tested implementation and hardware settings and therefore need not equal the asymptotic complexity exponents exactly.

Scaling with scale experts. The near-zero activation-memory scaling exponent for scale experts reported in Table 9 does not imply constant forward computation. SoftScale uses dense routing, in which every expert is evaluated before routing-weighted aggregation. Increasing the number of experts from 1 to 8 therefore increases inference time from 3.65 to 17.99 ms (4.93×) and model size from 1.33 to 7.95 MB (5.98×). Model size includes parameter and buffer storage but excludes activations, optimizer states, and temporary runtime memory. In comparison, incremental peak activation memory increases only from 217.19 to 218.97 MB (1.01×), because the experts are evaluated sequentially and their temporary buffers are reused. Total peak memory still increases slightly because the additional expert parameters and outputs must be retained.

E.5 Analysis of the number of scales

Figure 6 shows how the number of scales k affects forecasting performance across the four benchmarks. MSE exhibits a clear U-shaped trend on all datasets, with the lowest error consistently attained at $k = 3$: too few scales limit the range of temporal patterns that can be captured, while too many introduce redundant scales that dilute the routing signal. Beyond this shared pattern, the two extremes reveal dataset-specific sensitivities. On USHCN, the single-scale setting yields the worst MSE by a clear margin, consistent with the nature of climate data, where accurate 12-month-ahead forecasting benefits from jointly modeling multiple temporal regularities such as seasonal cycles and inter-annual trends. In contrast, MIMIC-III is the least tolerant of large k, with its MSE at $k = 5$ comparable to the single-scale setting. We attribute this to its high sparsity (96.7%) and large variable count (96): aggressively coarse scales smooth over the few valid observations per variable, obscuring fine-grained variations while offering little additional information beyond that captured at finer scales.

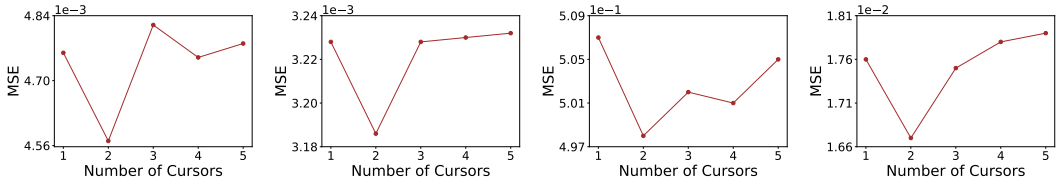


(a) PhysioNet (b) Activity (c) USHCN (d) MIMIC-III

Figure 6: Effect of different scale quantities on four datasets.

E.6 Analysis of cursor quantity

Figure 7 shows the effect of the number of cursors P on forecasting performance. The optimal value is consistently $P = 2$ across all four datasets, with MSE rising as P further increases. We interpret this as follows. $P = 1$ degenerates into a global pooling that ignores within-sample diversity and forfeits the benefit of soft segmentation. $P = 2$ is the minimal configuration that allows ACSS to separate two dominant sampling regimes—e.g., a densely-sampled acute phase versus a sparsely-sampled stable phase in clinical records, or distinct climate regimes within a 24-month window in USHCN—while the soft segmentation still lets transitional timestamps contribute to both cursors. Further increasing P distributes the limited observations over more cursors, reducing the effective observations per segment and increasing the risk of cursor redundancy, which is consistent with the motivation of our diversity loss $\mathcal{L}_{\text{dist}}$. These results indicate that a small number of cursors is generally sufficient on our benchmarks, and that excessive segmentation fragments the limited observations across cursors without yielding additional benefit for IMTS.

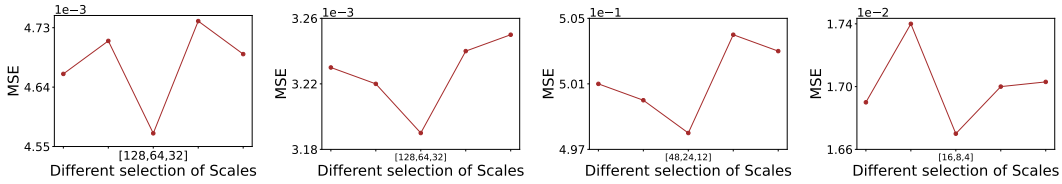


(a) Physionet (b) Human Activity (c) USHCN (d) MIMIC-III

Figure 7: Effect of different cursor quantities on four datasets.

E.7 Analysis of scale sizes

Figure 8 illustrates the impact of scale sizes on model performance. The optimal scale pools differ across datasets—$[48, 24, 12]$ for USHCN, $[128, 64, 32]$ for PhysioNet and Human Activity, and $[16, 8, 4]$ for MIMIC-III—reflecting that each dataset benefits from a different granularity range that best captures its underlying temporal patterns. For USHCN, which targets 12-month-ahead weather forecasting over a 24-month observation window, the largest scale in the pool is fine enough to resolve within-year variations, while the smallest still provides a coarse summary aligned with the annual cycle. In all four cases, MSE rises whenever the pool deviates from this sweet spot: using scales that are too large over-refines the representation and produces redundant tokens beyond what the data can support, while using scales that are too small sacrifices fine-grained local information.

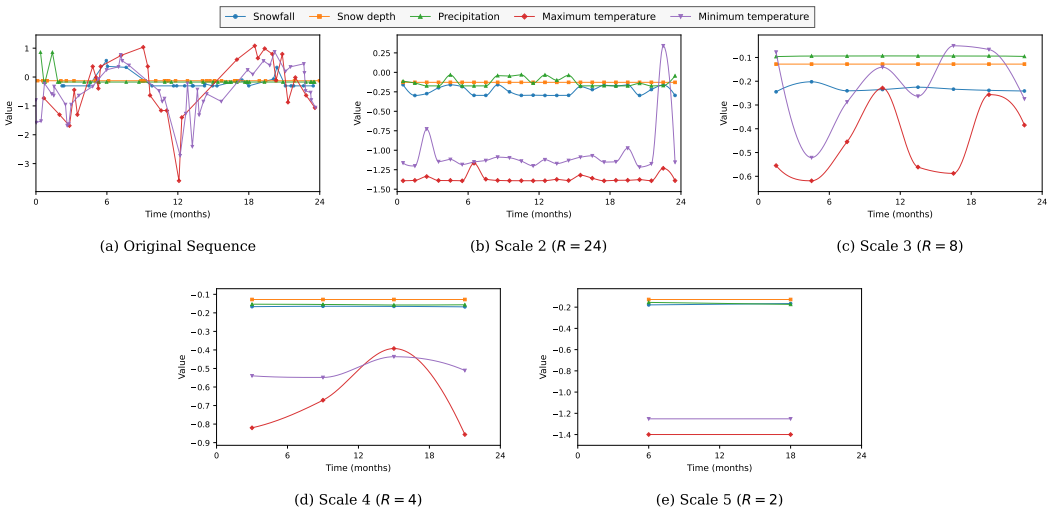


(a) Physionet (b) Human Activity (c) USHCN (d) MIMIC-III

Figure 8: Effect of different scale pool sizes on four datasets. For each dataset, we evaluate five scale pools around a central configuration (marked on the x-axis), where each neighboring pool is obtained by halving (to the left) or doubling (to the right) all three scale values. The central configuration corresponds to the best-performing pool for that dataset.

F VISUALIZATION OF MULTI-SCALE VIEWS

USHCN contains meteorological observations with recognizable climatic patterns. To examine whether the representations learned by SoftScale retain such patterns, we visualize a 24-month sequence together with its learned representations at four scales, $R \in \{24, 8, 4, 2\}$, in Figure 9. Here, R denotes the number of learned scale positions rather than a predefined temporal aggregation interval. These attention-based representations reveal climatic characteristics at progressively coarser scales.



(a) Original Sequence (b) Scale 2 ($R = 24$) (c) Scale 3 ($R = 8$)

(d) Scale 4 ($R = 4$) (e) Scale 5 ($R = 2$)

Figure 9: Visualization of a 24-month USHCN sequence and its learned representations at four scales. The learned views progressively summarize the original observations from fine-grained variations to broad annual-scale climatic characteristics.

Scale 1 (Original Sequence). The original observations retain complete temporal information, exhibiting substantial short-term variability together with recurring seasonal temperature changes.

Scale 2 ($R = 24$). The finest learned view preserves detailed local variations, particularly the short-term changes in maximum and minimum temperatures.

Scale 3 ($R = 8$). Transient fluctuations are reduced, while the synchronized rise and fall of maximum and minimum temperatures remain visible, making the recurring seasonal pattern clearer.

Scale 4 ($R = 4$). The coarser representation suppresses local fluctuations and emphasizes broad transitions between warmer and colder periods.

Scale 5 ($R = 2$). The coarsest view summarizes the 24-month sequence at an approximately annual scale, highlighting broad year-to-year climatic variations, such as overall warming, cooling, or relatively stable conditions across the two yearly periods.

Overall, the visualization shows that the learned multi-scale representations preserve meaningful temperature variations while progressively reducing short-term fluctuations. The resulting hierarchy ranges from detailed local changes to seasonal and annual-scale climatic characteristics, providing an intuitive interpretation of how the multi-scale experts capture distinct aspects of temporal patterns.