SCALE-AWARE FOUNDATION PRIORS FOR IRREGULAR MULTIVARIATE TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Irregular multivariate time series (IMTS) are widely encountered in sequential modeling, yet irregular sampling makes the effective temporal scale vary locally within a single series. However, most existing IMTS methods focus on representing asynchronous and missing observations, still lack explicit local scale adaptation, thereby overlooking this local scale heterogeneity. Foundation models already encode multi-scale temporal structure, but this prior remains locked to a regular sampling grid. To fill this gap, we propose ScaleIMTS, which adapts foundation priors to IMTS in a scale-aware manner. It first conditions these priors on local sampling scales by routing each temporal segment to scale-specific experts. The resulting representations are then refined by an IMTS-specific asynchronous structure: raw events are kept in time-aligned patches to capture local temporal patterns and time-varying inter-variable dependencies, and these features refine the already scale-adapted priors. The refined representations are fed into task-specific decoders to adapt to different downstream tasks. Experiments on 7 datasets demonstrate that ScaleIMTS outperforms state-of-the-art models in IMTS forecasting and classification tasks. Code is available at: `https://anonymous.4open.science/r/scale_imts-62E0`.

1 INTRODUCTION

With the proliferation of continuous monitoring systems, irregular multivariate time series (IMTS) are increasingly collected in healthcare, climate science, and human activity analysis (Shukla & Marlin, 2020; Sun et al., 2026; Bouchabou et al., 2021). Learning from IMTS is a fundamental sequential modeling problem, in which observations arrive as timestamped events that are asynchronous across variables and unevenly spaced in time. Irregular sampling therefore does more than induce missingness: it makes the effective temporal scale vary locally within a single series.

Figure 1 shows how this variation appears in one historical series. Along each variable, observations alternate between locally short and locally long intervals, which is visible both from the spacing of markers and from the interval strip beneath the curve. These short-scale and long-scale windows are staggered across variables, so a single timestamp can be fine-grained for one variable and coarse for another. The insets magnify two windows of v_1: in the short-scale window the latent oscillations are densely observed, whereas in the long-scale window only a few points remain on a slower trend. We refer to this sampling-dependent, asynchronous change of effective temporal granularity as *local scale heterogeneity*. The resulting problem is to learn representations that remain consistent as local sampling scales shift over time and across variables within a single input.

Recent IMTS methods have made substantial progress on representing asynchronous and missing observations. Alignment models absorb irregular timestamps into a shared temporal graph (Zhang et al., 2023; Chowdhury et al., 2023), while graph and patch architectures encode sparse events and cross-variable interactions (Yalavarthi et al., 2024; Zhang et al., 2024). Follow-up work further introduces hierarchical patches, hypergraphs, and recursive or token-mixing multi-scale encodings (Luo et al., 2025; Li et al., 2025; Zhong et al., 2025; Li et al., 2026). These models typically through a series-wide alignment, a predefined patch hierarchy, or a recursive scale schedule (Zhang et al., 2023; Luo et al., 2025; Li et al., 2026). Although they encode multiple resolutions, the scale schedule remains series-wide. Such multi-scale features therefore change resolution by construction, rather than tracking how the local inter-observation interval itself shifts over time and across variables,

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.

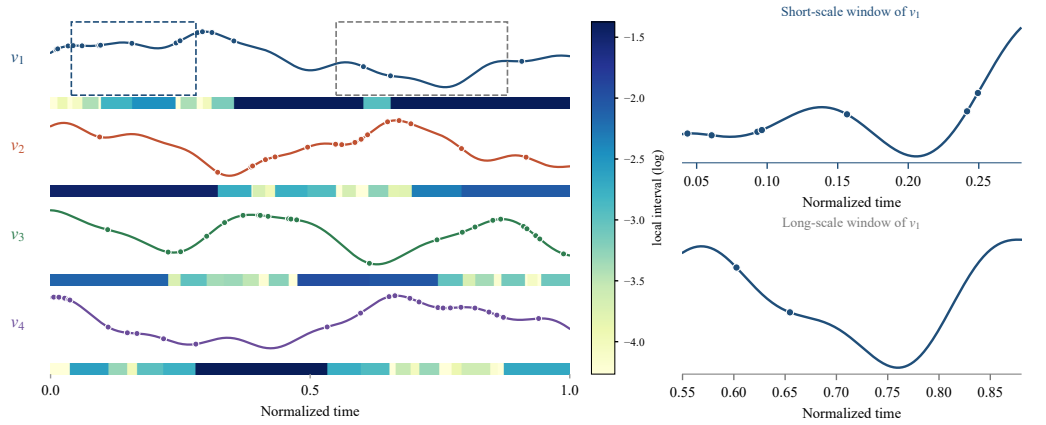


Figure 1: Local scale variation in one IMTS sample. Curves denote latent processes and markers denote observed events. The strip under each variable encodes the local inter-observation interval; darker indicates a longer local scale. Dashed boxes on v_1 mark a short-scale window and a long-scale window, which are magnified on the right.

so the sampling-induced local scale heterogeneity in Figure 1 remains unaddressed.In other words, they model multiple predefined scales of the series, not the local temporal scale that varies along the series.

Pretrained sequence models have been used as temporal backbones for time series analysis, and this line of work has been extended to irregular sampling (Zhou et al., 2023; Zhang et al., 2025). They encode temporal structure across a range of contexts, which matches local scale heterogeneity: short-scale windows need fine-grained dynamics, while long-scale windows need coarser temporal context. This prior, however, remains locked to a regular sampling grid (Zhou et al., 2023). Using it on IMTS therefore typically starts by mapping irregular events onto a series-wide aligned timeline so that the pretrained backbone can consume them (Zhang et al., 2025). The same alignment assigns every local region the same temporal structure, so short-scale and long-scale windows are processed alike, and it also places variables onto one timeline, so asynchronous observation times among variables are blurred. These challenges lead to unreliable local-pattern capture and degraded transfer of pretrained temporal structure.

To fill this gap, we propose ScaleIMTS, which adapts foundation priors to IMTS in a scale-aware manner. It first conditions these priors on local sampling scales by routing each temporal segment to scale-specific experts. The resulting representations are then refined by IMTS-specific asynchronous structure: raw events are kept in time-aligned patches to capture local temporal patterns and time-varying inter-variable dependencies, and these features refine the already scale-adapted priors. The refined representations are finally fed into task-specific decoders to adapt to different downstream tasks.

Our contributions are summarized as follows.

- We introduce a scale-aware adaptation of foundation priors that routes each temporal segment to scale-specific experts, so local sampling scales can vary over time and across variables.
- We refine the scale-adapted priors with inter-variable asynchrony, using event-level structure in time-aligned patches to inject asynchronous cross-variable relations into the pretrained representations.
- We evaluate ScaleIMTS on IMTS forecasting and classification, including settings with local scale perturbations.

2 Related Work

2.1 Irregular Multivariate Time Series

A central question in IMTS modeling is how asynchronously arriving observations should be represented. One line of work maps irregular timestamps onto a shared temporal structure, either by

time warping or by pre-training directly on irregular inputs (Zhang et al., 2023; Chowdhury et al., 2023). Another builds graphs or patches over observed events, so sparse measurements and cross-variable interactions can be encoded without first filling a dense grid (Yalavarthi et al., 2024; Zhang et al., 2024). Subsequent models organize these representations at several resolutions, including hierarchical patch graphs (Luo et al., 2025), hypergraph message passing (Li et al., 2025), adaptive spatio-temporal graph interactions (Liu et al., 2026a), and recursive or token-mixing multi-scale encodings (Zhong et al., 2025; Li et al., 2026). In each case the scale schedule is defined for the series as a whole—a shared alignment, a fixed patch hierarchy, a recursive split of the full horizon, or a graph constructed over the entire sample—and does not change with local inter-observation intervals or differ across variables at the same time.

2.2 Multi-scale Time Series Modeling

Multi-scale time series models extract features at several temporal resolutions that are specified in advance. For regularly sampled series, this is typically implemented by 2D variation modeling, fixed-length patches, hierarchical interpolation, adaptive pathways, or decomposable mixing across resolutions (Wu et al., 2023; Nie et al., 2023; Challu et al., 2023; Chen et al., 2024; Wang et al., 2024). Related constructions have been used for IMTS, including time warping, hierarchical patches, and recursive splits of the observation horizon (Zhang et al., 2023; Luo et al., 2025; Li et al., 2026). In both cases the scale schedule is shared across the series: it does not vary with local inter-observation intervals, and it is not defined separately for different variables at the same time.

2.3 Foundation models for time series

Pretrained sequence models have been adopted as temporal backbones for regularly sampled series (Zhou et al., 2023; Liu et al., 2024; Goswami et al., 2024). This approach has been extended to irregular sampling by encoding IMTS as token sequences for a pretrained model (Zhang et al., 2025). The prior is trained on a regular grid, so transfer typically maps events onto a series-wide aligned timeline; local short-scale and long-scale windows then share the same temporal structure, and asynchronous observation times among variables are blurred.

3 Problem Formulation

An IMTS sample is a set of observation events

$$\mathcal{X} = \{(t_i, x_i, c_i)\}_{i=1}^{N}, \tag{1}$$

where $t_i \in \mathbb{R}$ is the timestamp, $x_i \in \mathbb{R}$ is the observed value, and $c_i \in \{1, \dots, V\}$ is the variable index. Given a cutoff time t_s, the history is

$$\mathcal{X}_{\text{hist}} = \{(t_i, x_i, c_i) \in \mathcal{X} \mid t_i \le t_s\}. \tag{2}$$

Local scale is defined per variable and per time window. For variable v and window W, let $\Delta_{v,W}$ be the mean interval between consecutive observations of v in W. A small $\Delta_{v,W}$ corresponds to a short local scale and a large $\Delta_{v,W}$ to a long local scale. These windows need not align across variables, so the effective scale can differ over time and across variables within one sample. Forecasting is specified by a query set $\mathcal{Q} = \{(\tau_j, c_j)\}_{j=1}^{K}$ with $\tau_j > t_s$. The model predicts $\hat{\mathcal{Y}}_{\mathcal{Q}} = F_\theta(\mathcal{X}_{\text{hist}}, \mathcal{Q})$. Classification predicts a sample-level label $\hat{y} = F_\theta(\mathcal{X}_{\text{hist}})$. Both tasks share the representation of $\mathcal{X}_{\text{hist}}$.

4 Method

The overview of our proposed method, ScaleIMTS, is illustrated in Figure 2. ScaleIMTS adapts foundation priors to IMTS in a scale-aware manner. It first conditions pretrained temporal representations on local sampling scales by routing each temporal segment to scale-specific experts (Sec. 4.1). It then keeps raw events in time-aligned patches to capture local temporal patterns and inter-variable asynchrony (Sec. 4.2), and uses these features to refine the already scale-adapted priors (Sec. 4.3). The refined representations are fed into task-specific decoders for forecasting and classification.

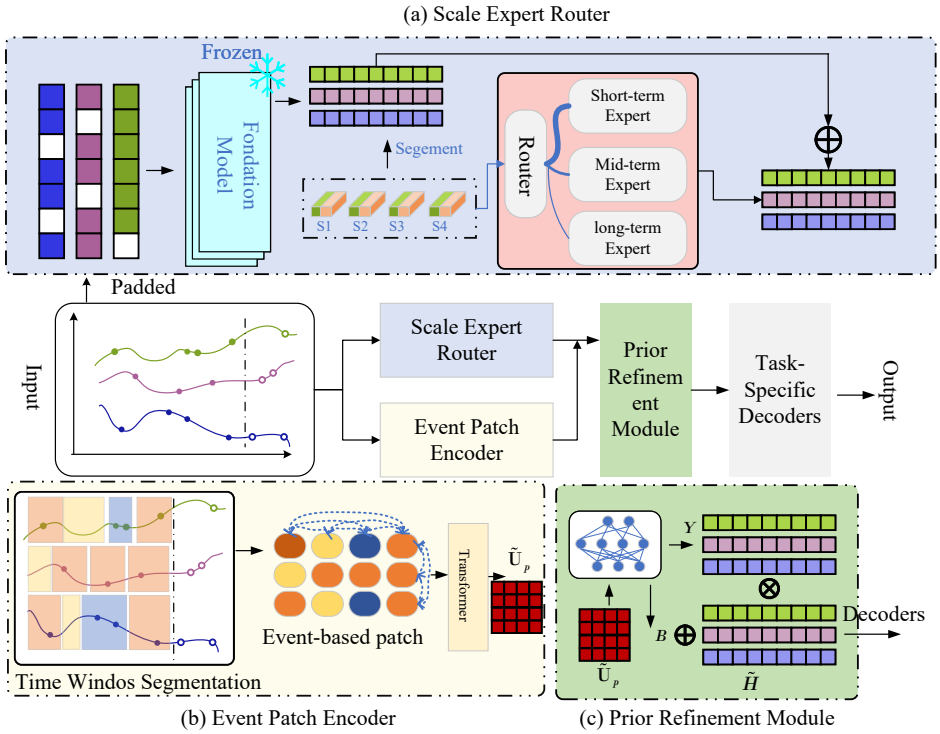


Figure 2: Overview of the ScaleIMTS architecture. The model adapts pretrained temporal representations to local sampling scales, models event-level irregular dynamics, and performs irregularity-aware foundation refinement for downstream forecasting and classification.

4.1 Scale Expert Router

The Scale Expert Router adapts a variable-wise pretrained temporal prior at the segment level, so short-scale and long-scale windows need not share the same temporal inductive bias. Batch indices are omitted below. We first turn $\mathcal{X}_{\text{hist}}$ into padded tensors $\mathbf{X} \in \mathbb{R}^{B\times L\times V}$, timestamps $\mathbf{T}$, and mask $\mathbf{M}$. Here L is the padded length: $x_{v,\ell}$ is the value of variable v at padded time index ℓ, t_ℓ is the timestamp of that index, and $m_{v,\ell} = 1$ if and only if variable v is observed there. All variables share $\mathbf{T}$; which events actually occur is marked by $\mathbf{M}$. We encode each position as $\mathbf{z}_{v,\ell} = \phi_{\text{val}}([x_{v,\ell}; m_{v,\ell}]) + m_{v,\ell}\phi_{\text{time}}(t_\ell)$, with time encodings applied only at observed entries. A pretrained temporal backbone then produces $\mathbf{H}_v^T = \mathcal{P}_T(\mathbf{Z}_v, \widetilde{\mathbf{m}}_v)$, where $\mathbf{Z}_v = [\mathbf{z}_{v,1}; \ldots; \mathbf{z}_{v,L}]$ and $\widetilde{\mathbf{m}}_v$ includes the prompt. Both pretrained encoders $\mathcal{P}_T$ and $\mathcal{P}_V$ are kept frozen.

We split $\mathbf{H}_v^T$ into S contiguous segments of equal token length, where S is a hyperparameter. The split follows the padded time index, and segments can span different amounts of time. Let $I_{v,s}$ be the valid token indices in segment s and let L_s be its padded token length. We define $\mathbf{r}_{v,s} = [\rho_{v,s}, \mu_{v,s}, \sigma_{v,s}, \delta_{v,s}]$, where $\rho_{v,s} = |I_{v,s}|/L_s$, and $\mu_{v,s}$, $\sigma_{v,s}$, and $\delta_{v,s} = \max_{\ell\in I_{v,s}} t_\ell - \min_{\ell\in I_{v,s}} t_\ell$ are respectively the mean, standard deviation, and span of the valid timestamps. These statistics provide a robust proxy for the local sampling scale $\Delta_{v,W}$; all four are set to zero for an empty segment.

Together with the masked pooled prior $\bar{\mathbf{h}}_{v,s}$, the scale vector $\mathbf{d}_{v,s} = \text{LN}(\phi_{\text{feat}}(\bar{\mathbf{h}}_{v,s}) + \phi_{\text{stat}}(\mathbf{r}_{v,s}))$ summarizes how densely the segment is observed, where ϕ_{feat} and ϕ_{stat} are learnable maps. A router produces soft weights over short-, mid-, and long-context experts:

$$\boldsymbol{\pi}_{v,s} = \text{Softmax}_k\left(\frac{\mathcal{R}\left([\mathbf{d}_{v,s}; \mathbf{q}; \phi_I(\mathbf{g}^I); \mathbf{p}_v]\right)}{\tau}\right), \tag{3}$$

where $k \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$ indexes the three experts, $\pi_{v,s}^{(k)}$ is the corresponding mixture weight, and $\tau > 0$ controls the routing sharpness. The query context $\mathbf{q}$ is the mean of the encoded future-query vectors; for classification, it encodes the last valid timestamp. The vector $\mathbf{g}^I$ is the global irregular context from Sec. 4.2.

All segment summaries are computed over valid tokens only. Specifically, $\mathbf{h}_{v,s}^{\mathrm{last}}$ and $\mathbf{h}_{v,s}^{\mathrm{first}}$ are the last and first valid temporal tokens, $\mathbf{h}_{v,s}^{\mathrm{std}}$ is their element-wise standard deviation, and $\mathbf{h}_{v,s}^{\mathrm{max}}$ is their element-wise maximum; all summaries are zero for an empty segment. We use three independent expert MLPs with different summary inputs,

$$\begin{aligned}
\mathbf{e}_{v,s}^{\mathrm{S}} &= \mathcal{E}_{\mathrm{S}}\left([\mathbf{d}_{v,s}; \mathbf{h}_{v,s}^{\mathrm{last}}; \bar{\mathbf{h}}_{v,s}; \mathbf{r}_{v,s}]\right), \\
\mathbf{e}_{v,s}^{\mathrm{M}} &= \mathcal{E}_{\mathrm{M}}\left([\mathbf{d}_{v,s}; \bar{\mathbf{h}}_{v,s}; \mathbf{h}_{v,s}^{\mathrm{std}}; \mathbf{r}_{v,s}]\right), \\
\mathbf{e}_{v,s}^{\mathrm{L}} &= \mathcal{E}_{\mathrm{L}}\left([\mathbf{d}_{v,s}; \bar{\mathbf{h}}_{v,s}; \mathbf{h}_{v,s}^{\mathrm{max}}; \mathbf{h}_{v,s}^{\mathrm{first}}; \mathbf{r}_{v,s}]\right),
\end{aligned} \tag{4}$$

where $\mathcal{E}_{\mathrm{S}}$, $\mathcal{E}_{\mathrm{M}}$, and $\mathcal{E}_{\mathrm{L}}$ have independent parameters and consist of a two-layer MLP. The superscripts S, M, and L identify the short-, mid-, and long-context experts. The corresponding vectors $\mathbf{e}_{v,s}^{\mathrm{S}}$, $\mathbf{e}_{v,s}^{\mathrm{M}}$, and $\mathbf{e}_{v,s}^{\mathrm{L}}$ are expert-generated corrections to the pretrained segment representation, rather than probabilities or token states. Their different input summaries impose complementary inductive biases: the short expert emphasizes recency, the mid expert emphasizes within-segment dispersion, and the long expert uses segment-wide boundary and maximum responses.

The expert outputs are softly mixed as

$$\Delta\mathbf{h}_{v,s} = \sum_k \pi_{v,s}^{(k)} \mathbf{e}_{v,s}^{(k)}, \tag{5}$$

where $\pi_{v,s}^{(k)}$ is produced by the router and $\mathbf{e}_{v,s}^{(k)}$ is the corresponding expert output defined above. Since $\Delta\mathbf{h}_{v,s}$ summarizes the local sampling regime of the entire segment, we broadcast it to the valid temporal tokens belonging to that segment. Let $s(\ell)$ be the segment assignment induced by the equal-length partition above. The adapted token representation is

$$\widetilde{\mathbf{h}}_{v,\ell}^T = \mathbf{h}_{v,\ell}^T + \alpha\Delta\mathbf{h}_{v,s(\ell)}, \qquad \alpha = \alpha_{\max}\,\mathrm{sigmoid}(a). \tag{6}$$

Here a is a learnable residual-gate logit, and the sigmoid converts it into a multiplicative gate whose maximum strength is controlled by the hyperparameter $\alpha_{\max}$. The adapted tokens $\widetilde{\mathbf{h}}_{v,\ell}^T$ are pooled over valid positions and passed through a pretrained variable encoder $\mathcal{P}_V$. We write $\mathbf{f}_v$ for the resulting variable-level prior, which is then linearly mapped to the hidden size used in Sec. 4.3.

4.2 Event Patch Encoder

Although Sec. 4.1 encodes timestamps at observed positions, its padded token sequence does not explicitly model variable-size event sets or patch-specific cross-variable interactions. The Event Patch Encoder complements it by organizing raw observations into time-aligned event patches and modeling within-patch intervals and time-varying inter-variable dependencies. Batch indices are omitted as in Sec. 4.1.

Let P be the patch length in time units and $T_{\max}$ the time span of the series. Valid observations are assigned to n time-aligned patches of width $P/T_{\max}$. For variable v and patch p, the event set $\mathcal{E}_{v,p}$ keeps observed values and timestamps in temporal order.

Each event r is encoded as $\mathbf{e}_{v,p,r}$ from its value, timestamp, and interval to the previous event (zero for the first event). Content-adaptive logits $o_{v,p,r}$ yield weights $\alpha_{v,p,r} = \mathrm{Softmax}_r(o_{v,p,r})$, which aggregate the variable-length set into a patch representation

$$\mathbf{h}_{v,p} = \mathrm{LN}\left(\mathrm{ReLU}\left(\sum_r \alpha_{v,p,r}\mathbf{e}_{v,p,r} + \mathbf{b}\right)\right), \tag{7}$$

where $\mathbf{b}$ is a learnable bias.

The patch tokens of each variable, $\mathbf{H}_v^P = [\mathbf{h}_{v,1}; \ldots; \mathbf{h}_{v,n}]$, form an ordered sequence. A Transformer encoder, with sinusoidal positions $\mathbf{E}$ and patch-valid mask $\mathbf{M}_v^P$, models dependencies across patches:

$$\mathbf{U}_v = \mathrm{LN}\left(\mathrm{TransformerEncoder}(\mathbf{H}_v^P + \mathbf{E}; \mathbf{M}_v^P)\right). \tag{8}$$

Cross-variable relations may still change from patch to patch. For patch p, let $\mathbf{U}_p \in \mathbb{R}^{V \times D}$ collect all variables, where D is the hidden size. Learnable maps $\mathbf{W}_Q$ and $\mathbf{W}_K$ give $\mathbf{Q}_p = \mathbf{U}_p\mathbf{W}_Q$ and $\mathbf{K}_p = \mathbf{U}_p\mathbf{W}_K$, and induce a directed interaction graph

$$\mathbf{A}_p = \mathrm{Softmax}\left(\mathrm{ReLU}\left(\mathbf{Q}_p\mathbf{K}_p^\top\right)\right), \tag{9}$$

which is applied through a residual MLP ψ,

$$\widetilde{\mathbf{U}}_p = \mathrm{LN}\left(\mathbf{U}_p + \psi\left([\mathbf{U}_p; \mathbf{A}_p\mathbf{U}_p]\right)\right). \tag{10}$$

Pooling over valid patches yields variable-level features for Sec. 4.3; pooling over valid variables yields the global context $\mathbf{g}^I$ used in Sec. 4.1.

4.3 Prior Refinement Module

The variable-level prior $\mathbf{h}_v$ from Sec. 4.1 no longer carries the patch-level cross-variable structure of Sec. 4.2. Let $\mathbf{u}_v$ be the variable-level feature obtained by pooling $\widetilde{\mathbf{U}}_p$ over valid patches.

A query vector $\mathbf{q}_\tau$ encodes the target time τ (the last valid timestamp for classification). Learnable maps send $\mathbf{h}_v$, $\mathbf{u}_v$, and $\mathbf{q}_\tau$ to a shared space. An MLP g then produces a per-variable, per-query affine modulation from the asynchronous feature and the query,

$$\boldsymbol{\gamma}_{v,\tau},\ \boldsymbol{\beta}_{v,\tau} = g([\mathbf{u}_v; \mathbf{q}_\tau]). \tag{11}$$

The scale-adapted prior is updated as

$$\mathbf{z}_{v,\tau} = [(1 + \boldsymbol{\gamma}_{v,\tau}) \odot \mathbf{h}_v + \boldsymbol{\beta}_{v,\tau}, \mathbf{q}_\tau]. \tag{12}$$

The refined feature is passed to the task-specific decoders in Appendix C.

5 Experiments

5.1 Experimental Setting

We evaluate forecasting and classification on 7 IMTS datasets. Forecasting uses PhysioNet (Silva et al., 2012), MIMIC-III (Johnson et al., 2016), Human Activity (Kaluža et al., 2010), and USHCN (Menne et al., 2015), with the preprocessing of t-PatchGNN (Zhang et al., 2024). Classification uses P12 (Goldberger et al., 2000), P19 (Reyna et al., 2019), and PAM (Reiss & Stricker, 2012) with the official split files of Raindrop (Zhang et al., 2022) and ISTS-PLM (Zhang et al., 2025). Dataset statistics are given in Appendix B.

Baselines are IMTS models and a pretrained IMTS backbone. The IMTS group includes GRU-D (Che et al., 2018), SeFT (Horn et al., 2020), mTAN (Shukla & Marlin, 2021), CRU (Schirmer et al., 2022), PrimeNet (Chowdhury et al., 2023), Raindrop (Zhang et al., 2022), Warpformer (Zhang et al., 2023), t-PatchGNN (Zhang et al., 2024), GraFITi (Yalavarthi et al., 2024), APN (Liu et al., 2026b), Hi-Patch (Luo et al., 2025), HyperIMTS (Li et al., 2025), ReIMTS (Li et al., 2026), and KAFNet (Zhou et al., 2026). The pretrained backbone is ISTS-PLM (Zhang et al., 2025). Classification compares against GRU-D (Che et al., 2018), SeFT (Horn et al., 2020), mTAN (Shukla & Marlin, 2021), Raindrop (Zhang et al., 2022), Warpformer (Zhang et al., 2023), and Hi-Patch (Luo et al., 2025), which share the Raindrop/ISTS-PLM official-split protocol. Implementations follow the original papers; hyperparameters are in Appendix D.

For forecasting, each dataset is split into training, validation, and test sets with ratio 6:2:2. We report MSE and MAE on the default horizon of each dataset; additional observation and forecast windows are in the appendix. For classification, P12, P19, and PAM use the five splits by Raindrop. We report AUROC and AUPRC on P12 and P19, and accuracy, precision, recall, and F1 on PAM.

Training uses Adam with early stopping on the validation loss, and each setting is repeated over 5 random seeds. The pretrained encoders $\mathcal{P}_T$ and $\mathcal{P}_V$ are frozen; the input embeddings, variable prompts, scale router, experts, and remaining adaptation modules are trained. The scale router uses $S = 4$ segments and three experts. Further training details are in Appendix F.1.

Table 1: Forecasting on the default horizon: Activity 3000→1000 ms, USHCN 24→1 month, PhysioNet 24→24 h, MIMIC-III 24→24 h. Results are MSE/MAE, mean±std over five seeds. Best in **bold**, second-best underlined among methods.

	Activity		USHCN		PhysioNet		MIMIC-III	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
GRU-D	0.0887±0.0113	0.2079±0.0176	0.3677±0.1014	0.3588±0.0287	0.5325±0.1550	0.5315±0.0928	0.4900±0.0180	0.4392±0.0095
SeFT	0.1886±0.1079	0.3043±0.0950	0.2993±0.0910	0.3201±0.0218	0.7611±0.0017	0.6595±0.0021	0.6219±0.0081	0.5091±0.0061
mTAN	0.3051±0.0013	0.4057±0.0015	0.3660±0.1100	0.3456±0.0287	0.7612±0.0015	0.6600±0.0018	0.6327±0.0178	0.5240±0.0097
CRU	0.2959±0.0229	0.4046±0.0067	0.3324±0.1161	0.3317±0.0300	0.4952±0.0055	0.5072±0.0051	0.6350±0.0098	0.5169±0.0052
PrimeNet	0.3060±0.0014	0.4071±0.0022	0.4271±0.0821	0.3926±0.0143	0.7621±0.0014	0.6603±0.0020	0.7294±0.0532	0.5652±0.0231
Raindrop	0.3051±0.0012	0.4060±0.0026	0.3797±0.1265	0.3504±0.0354	0.7610±0.0015	0.6587±0.0016	0.9687±0.0006	0.6593±0.0005
Warpformer	0.3002±0.0116	0.4042±0.0070	0.3388±0.1440	0.3230±0.0590	0.7650±0.0022	0.6694±0.0042	0.5568±0.0138	0.4740±0.0105
t-PatchGNN	0.2204±0.0714	0.3287±0.0641	0.4286±0.1344	0.3749±0.0711	0.6861±0.1534	0.6144±0.0969	0.5482±0.0194	0.4675±0.0118
GraFITi	0.0586±0.0016	0.1474±0.0024	0.2619±0.0410	0.3059±0.0071	0.3017±0.0015	0.3669±0.0011	0.4806±0.0112	0.4208±0.0064
APN	0.0432±0.0008	0.1188±0.0023	0.2916±0.0747	0.3022±0.0213	0.3103±0.0022	0.3773±0.0047	0.4868±0.0126	0.4248±0.0071
Hi-Patch	0.8963±0.6627	0.7101±0.3721	0.3688±0.1403	0.3443±0.0321	0.5844±0.2450	0.5520±0.1552	0.4778±0.0119	0.4151±0.0031
HyperIMTS	0.0461±0.0003	0.1297±0.0009	0.3140±0.0862	0.3211±0.0256	**0.2972±0.0018**	0.3673±0.0026	0.4851±0.0136	0.4239±0.0075
ReIMTS	0.0436±0.0008	0.1209±0.0034	0.3092±0.0868	0.3304±0.0163	0.3060±0.0042	0.3737±0.0054	0.4884±0.0151	0.4266±0.0092
KAFNet	0.0432±0.0004	0.1173±0.0024	0.3386±0.1012	0.3273±0.0220	0.4093±0.1978	0.4385±0.1266	0.4667±0.0142	0.4187±0.0111
ISTS-PLM	0.0746±0.0016	0.1779±0.0048	0.3266±0.0695	0.3492±0.0391	0.4408±0.1634	0.4644±0.1028	0.5207±0.0178	0.4513±0.0106
ScaleIMTS	**0.0425±0.0003**	**0.1154±0.0012**	**0.2157±0.0049**	**0.2890±0.0033**	0.2988±0.0014	**0.3656±0.0019**	**0.4255±0.0081**	**0.3871±0.0025**

Table 2: Classification on the 5 splits by Raindrop. Results are mean ± std over five seeds. Best in **bold**, second-best underlined. AUROC and AUPRC are percentages.

	P12		P19		PAM			
Method	AUROC	AUPRC	AUROC	AUPRC	Acc.	Prec.	Rec.	F1
GRU-D	84.8±0.7	49.7±1.0	92.9±0.3	67.8±0.4	96.4±1.0	96.7±1.1	96.8±0.8	96.7±0.9
SeFT	84.6±0.7	48.4±1.2	90.1±0.5	62.4±1.1	94.6±1.2	95.4±1.2	95.2±1.0	95.3±1.1
mTAN	86.2±0.3	51.9±0.9	82.6±1.4	33.4±3.0	95.4±0.7	95.7±0.7	95.9±0.5	95.8±0.6
Raindrop	77.0±3.7	34.8±2.1	81.1±1.6	47.3±1.4	84.8±0.4	87.6±1.0	86.3±0.3	86.8±0.4
Warpformer	86.2±0.2	51.8±0.3	92.3±0.8	65.4±1.9	95.3±0.4	96.6±0.5	95.6±0.5	96.0±0.5
Hi-Patch	84.9±0.6	47.4±1.6	91.3±1.0	60.1±3.0	96.6±1.1	97.4±1.0	96.9±1.0	97.1±1.0
ScaleIMTS	**86.5±0.6**	**52.6±0.9**	**93.4±0.2**	**68.2±0.3**	**97.3±0.1**	**97.9±0.1**	**97.3±0.1**	**97.6±0.1**

5.2 Main Results

Table 1 reports forecasting results on the default horizon of each dataset, and Table 2 reports classification scores. Additional observation and forecast windows are in Appendix G.3.

ScaleIMTS attains the best mean MAE on all four forecasting datasets and the best mean MSE on Human Activity, USHCN, and MIMIC-III. The improvements over recent IMTS methods are most evident on USHCN and MIMIC-III, where ScaleIMTS reduces MSE from 0.2619 (GraFITi) to 0.2157 and from 0.4667 (KAFNet) to 0.4255. On Human Activity, it further improves upon the already strong results of APN and KAFNet. On PhysioNet, ScaleIMTS records the best MAE (0.3656) while remaining competitive in MSE with the strongest hypergraph baseline. Relative to ISTS-PLM, which transfers the same frozen temporal prior on an aligned token grid, ScaleIMTS lowers both MSE and MAE on every dataset. These results indicate that conditioning the pretrained prior on local sampling scales, and refining it with event-level asynchronous structure, yields more accurate IMTS forecasts than using a series-wide scale schedule or an aligned-grid transfer of the prior.

P12 and P19 are imbalanced binary tasks, so we report AUROC and AUPRC; PAM is an eight-class activity task, so we report accuracy, precision, recall, and F1. ScaleIMTS achieves the best mean on every metric in Table 2. It improves both AUROC and AUPRC on the two clinical datasets, and accuracy, precision, recall, and F1 on PAM. The same scale-adapted representation therefore transfers from forecasting to classification, including settings in which the label depends on short activity patterns.

Table 3: Single-seed ablation of ScaleIMTS on Human Activity, USHCN, P12, and P19. Forecasting reports MSE/MAE; AUROC and AUPRC are percentages. Best in **bold**, second-best underlined.

	Activity		USHCN		P12		P19	
Variant	MSE	MAE	MSE	MAE	AUROC	AUPRC	AUROC	AUPRC
Full	**0.0426**	**0.1147**	**0.2172**	**0.2909**	86.7	**52.9**	**93.6**	**68.1**
w/o Local Scale	0.0491	0.1357	0.2964	0.2998	85.8	52.4	90.7	65.2
Global Routing	0.0500	0.1315	0.3028	0.3084	86.2	52.4	91.8	66.0
w/o Scale Statistics	0.0480	0.1319	0.3020	0.3433	**86.9**	52.2	91.1	66.2
w/o Event Patch	0.0476	0.1328	0.2732	0.3698	85.8	52.0	91.5	64.3
w/o Foundation Prior	1.3817	0.9785	0.6918	0.5140	63.9	22.0	85.2	51.0
Additive Update	0.0474	0.1349	0.2853	0.2974	85.9	51.3	91.0	65.2
Concat Fusion	0.3104	0.4112	0.4913	0.4051	86.2	51.9	90.4	62.4

5.3 Ablation Study

We next examine the contribution of each component on Human Activity, USHCN, P12, and P19. Table 3 groups the variants by local scale routing, asynchronous structure, and prior refinement.

For routing, **w/o Local Scale** disables the segment-level expert residual; **Global Routing** sets $S = 1$, so the whole series shares one scale; **w/o Scale Statistics** removes $\mathbf{r}_{v,s}$ from the router and experts. For asynchronous structure, **w/o Event Patch** removes the Event Patch Encoder of Sec. 4.2, and **w/o Foundation Prior** removes the pretrained temporal prior and the Scale Expert Router. For refinement, **Additive Update** replaces the affine rewrite of Sec. 4.3 with $\mathbf{f}_v + \mathbf{u}_v + \mathbf{q}_\tau$, and **Concat Fusion** concatenates the two features.

The full model is strongest overall. Removing the frozen temporal prior leads to a sharp drop on both tasks, which confirms that ScaleIMTS benefits from adapting foundation representations rather than learning temporal structure from scratch. Disabling segment-level experts, collapsing routing to a single global scale, or dropping the sampling-scale statistics increases forecasting error on Activity and USHCN and lowers P19 AUROC, supporting local, statistic-aware routing. Removing the Event Patch Encoder raises USHCN MAE from 0.2909 to 0.3698 and reduces AUPRC on both clinical datasets, consistent with the role of asynchronous event structure. Replacing the affine rewrite with addition or concatenation also underperforms the full model, indicating that the pretrained prior is better preserved by a query-conditioned modulation than by a direct merge of the two branches.

5.4 Parameter Study

The remaining routing hyperparameters are the number of segments S and the router temperature τ. We keep three experts and the rest of ScaleIMTS fixed, and evaluate $S \in \{1, 2, 4, 8\}$ at $\tau = 1$ and $\tau \in \{0.25, 0.5, 1, 2\}$ at $S = 4$ on the same four datasets as the ablation study. The default $(S, \tau) = (4, 1)$ is shared by both scans. Each configuration uses a single seed. Training and validation remain unperturbed; the dashed curves in Figure 3 evaluate the selected checkpoint on a test set with 30% local-scale perturbation.

Figure 3 plots both metrics against S. $S = 1$ assigns one scale to the whole series; larger S partitions the token sequence into shorter segments. The gray band marks the default $S = 4$ used in the main experiments. Relative to a series-wide scale ($S = 1$), segment-wise routing reduces MSE and MAE on both USHCN and Human Activity, on the clean test set and under the 30% local-scale perturbation. Among the scanned values, $S = 4$ attains the lowest clean MSE and MAE on both forecasting datasets and the lowest perturbed MSE; on the classification tasks it attains the highest clean AUROC on P12 and the highest clean AUROC and AUPRC on P19. A complementary scan over τ is reported in Appendix G.1.

5.5 Local Scale Perturbation

To assess the stability of learned representations under local changes in sampling density, we train each model on the original data and apply a controlled perturbation only to the test set. Specifically, a prescribed fraction of observations is preferentially removed from one temporal subregion, pro-

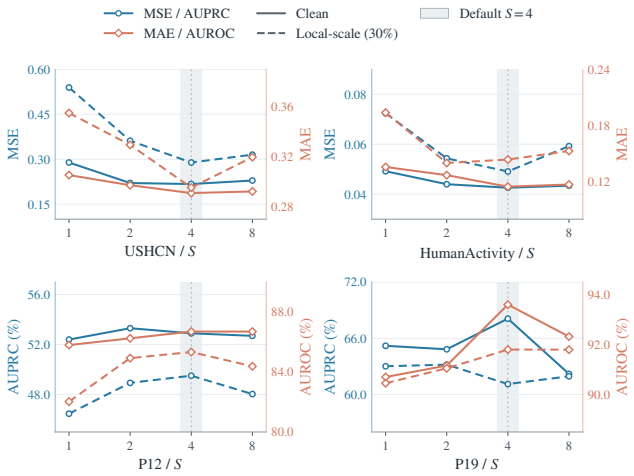


Figure 3: Sensitivity to the number of segments S at $\tau = 1$. Top: USHCN and Human Activity. Bottom: P12 and P19. Forecasting uses MSE (left axis, blue circles) and MAE (right axis, orange diamonds). Classification uses AUPRC (left) and AUROC (right), in percent. Solid lines are the clean test set; dashed lines use 30% local scale perturbation at test time. The x-axis is $\log_2 S$. The gray band marks the default $S = 4$.

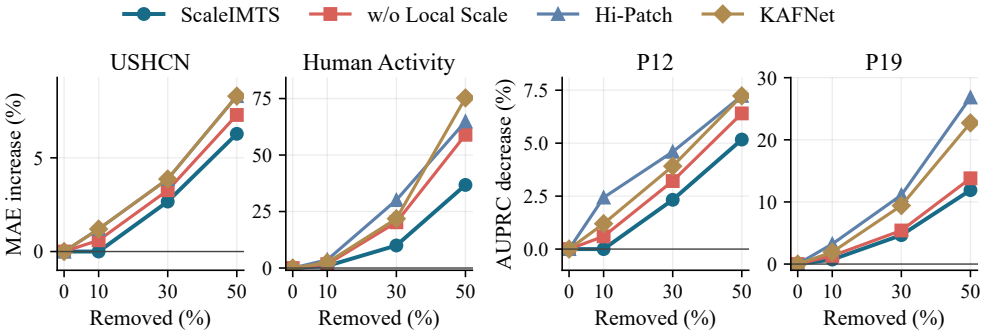


Figure 4: Relative degradation under local scale perturbation. Left: USHCN and Human Activity (MAE increase). Right: P12 and P19 (AUPRC decrease). The x-axis is the fraction of observations removed in selected windows.

ducing a localized transition from denser to sparser sampling while preserving the original time axis and prediction targets. We compare ScaleIMTS with **w/o Local Scale**, Hi-Patch, and KAFNet.

Figure 4 reports the relative MAE increase (forecasting) and AUPRC decrease (classification) as this local scale removal rate grows from 0% to 50%. As the perturbation intensifies, ScaleIMTS remains the most stable of the four methods: at 50% removal it records the smallest MAE increase on USHCN and Human Activity, and the smallest AUPRC decrease on P12 and P19. Segment-wise routing is most helpful in this high-perturbation regime, where the full model stays below Hi-Patch and KAFNet on every dataset. Random deletion and block missing are compared in Appendix G.2.

6 Conclusion

Irregular sampling makes the temporal scale vary locally within a series and asynchronously across variables, and this local scale heterogeneity is not handled by a series-wide scale schedule. ScaleIMTS refine a frozen pretrained temporal prior to that variation: a router, conditioned on a local scale vector, routes each token segment to short-, mid-, and long-context experts; an event-patch encoder models raw timestamps and time-varying cross-variable structure; and an affine rewrite injects those features into the already scale-adapted prior. On seven IMTS datasets for forecasting and classification, this adaptation outperforms IMTS baselines and a pretrained backbone.

AI USE STATEMENT

In this work, we used generative AI tools for implement methods and assist with translation. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, interpret results and generate synthetic data sets and clean and reformat dataset are not applicable to this work. Additionally, we used generative AI tools for edit a research paper to improve readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide an anonymized implementation of ScaleIMTS at `https://anonymous.4open.science/r/scale_imts-62E0`. Dataset descriptions, preprocessing, and evaluation settings are provided in Appendix B; additional experimental analyses are reported in Appendix G; the inference procedure is summarized in Appendix A; and the complete optimization objectives and implementation details are provided in Appendix F. The released code includes the configurations and scripts required to reproduce the main experiments.

REFERENCES

Damien Bouchabou, Sao Mai Nguyen, Christophe Lohr, Benoit LeDuc, and Ioannis Kanellos. A survey of human activity recognition in smart homes based on IoT sensors algorithms: Taxonomies, challenges, and opportunities with deep learning. *Sensors*, 21(18):6037, 2021. doi: 10.3390/s21186037.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018. doi: 10.1038/s41598-018-24271-9.

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 *International Conference on Learning Representations*, 2024.

Ranak Roy Chowdhury, Jiacheng Li, Xiyuan Zhang, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. PrimeNet: Pre-training for irregular multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. doi: 10.1609/aaai.v37i6.25876.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Wang, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, 2020.

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):160035, 2016.

Boštjan Kaluža, Violeta Mirchevska, Erik Dovgan, Mitja Luštrek, and Matjaž Gams. An agent-based approach to care in independent living. In *International joint conference on ambient intelligence*, pp. 177–186. Springer, 2010.

Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. HyperIMTS: Hypergraph neural network for irregular multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Boyuan Li, Zhen Liu, Yicheng Luo, and Qianli Ma. Learning recursive multi-scale representations for irregular multivariate time series forecasting. In *International Conference on Learning Representations*, 2026.

Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. In *International Conference on Learning Representations*, 2026a.

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 *AAAI Conference on Artificial Intelligence*, 2026b.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Yanson Luo et al. Hi-patch: Hierarchical patch GNN for irregular multivariate time series. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. OpenReview: nBgQ66iEUu.

MJ Menne, C Williams Jr, R 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 *International Conference on Learning Representations*, 2023.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pp. 108–109. IEEE, 2012. doi: 10.1109/iswc.2012.13. URL http://dx.doi.org/10.1109/iswc.2012.13.

Matthew A Reyna, Chris Josef, Salman Seyedi, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D Clifford. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In *2019 Computing in Cardiology (CinC)*, pp. Page–1. IEEE, 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.

Satya Narayan Shukla and Benjamin M. Marlin. A survey on principles, models and methods for learning from irregularly sampled time series: From discretization to attention and invariance. In *NeurIPS Workshop on ML Retrospectives, Surveys & Meta-Analyses*, 2020. URL https://ml-retrospectives.github.io/neurips2020/camera_ready/16.pdf.

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

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.

Chenxi Sun, Moxian Song, Derun Cai, Baofeng Zhang, Hongyan Li, and Shenda Hong. A review of deep learning methods for irregularly sampled medical time series data. *Health Data Science*, 6:0456, 2026. doi: 10.34133/hds.0456. URL https://spj.science.org/doi/10.34133/hds.0456.

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 *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 *International Conference on Learning Representations*, 2023.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. GraFITi: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i15.29560.

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*, 2023. doi: 10.1145/3580305.3599543.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Weijia Zhang, Chenlong Yin, Hao Liu, and Hui Xiong. Unleashing the power of pre-trained language models for irregularly sampled time series. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2025. doi: 10.1145/3711896.3737171.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022.

Shuhan Zhong, Weipeng Zhuo, Sizhe Song, Guanyao Li, Zhongyi Yu, and S.-H. Gary Chan. MTM: A multi-scale token mixing transformer for irregular multivariate time series classification. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2025. doi: 10.1145/3711896.3737058.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ziyu Zhou, Yiming Huang, Yanyun Wang, Yuankai Wu, James Kwok, and Yuxuan Liang. Revitalizing canonical pre-alignment for irregular multivariate time series forecasting. In *AAAI Conference on Artificial Intelligence*, 2026.

A Algorithm of ScaleIMTS

Algorithm 1 summarizes the forward pass. The Event Patch Encoder is computed first so that the global context $\mathbf{g}^I$ is available to the scale router, as in the implementation. Batch indices are omitted, and $\mathcal{P}_T$ and $\mathcal{P}_V$ remain frozen.

B Dataset Details

We report results on 7 IMTS datasets, four for forecasting and three for classification. Tables 4 and 5 list sample counts, numbers of variables, average sequence lengths, and missing ratios.

B.1 Forecasting

Forecasting uses Human Activity, USHCN, PhysioNet, and MIMIC-III, following the preprocessing of t-PatchGNN (Zhang et al., 2024). Each dataset is evaluated at three observation/forecast windows, described below.

Algorithm 1 Forward pass of ScaleIMTS

Require: history $\mathcal{X}_{\text{hist}}$; queries $\mathcal{Q}$ (last valid timestamp for classification); segments S; patch length P

Ensure: forecasts $\{\hat{y}_{v,\tau}\}$ or class scores $\hat{y}$

Pad $\mathcal{X}_{\text{hist}}$ to $\mathbf{X}, \mathbf{T}, \mathbf{M}$
Encode each query time as $\mathbf{q}_\tau$; $\mathbf{q} \leftarrow \text{mean}_\tau(\mathbf{q}_\tau)$
Assign valid events to n time-aligned patches of width $P/T_{\max}$
for $v = 1$ to V **do**
  for $p = 1$ to n **do**
    Encode events in $\mathcal{E}_{v,p}$ and aggregate to $\mathbf{h}_{v,p}$
  end for
  $\mathbf{U}_v \leftarrow \text{LN}(\text{TransformerEncoder}(\mathbf{H}_v^P + \mathbf{E}; \mathbf{M}_v^P))$
end for
for $p = 1$ to n **do**
  $\mathbf{A}_p \leftarrow \text{Softmax}(\text{ReLU}(\mathbf{Q}_p \mathbf{K}_p^\top))$
  $\widetilde{\mathbf{U}}_p \leftarrow \text{LN}(\mathbf{U}_p + \psi([\mathbf{U}_p; \mathbf{A}_p \mathbf{U}_p]))$
end for
Pool $\widetilde{\mathbf{U}}_p$ to $\mathbf{u}_v$ and $\mathbf{g}^I$
for $v = 1$ to V **do**
  $\mathbf{z}_{v,\ell} \leftarrow \phi_{\text{val}}([x_{v,\ell}; m_{v,\ell}]) + m_{v,\ell}\phi_{\text{time}}(t_\ell), \quad \mathbf{Z}_v \leftarrow [\mathbf{p}_v; \mathbf{z}_{v,1}; \ldots; \mathbf{z}_{v,L}]$
  $\mathbf{H}_v^T \leftarrow \mathcal{P}_T(\mathbf{Z}_v, \widetilde{\mathbf{m}}_v)$
  Split $\mathbf{H}_v^T$ into S equal-token segments
  for $s = 1$ to S **do**
    $\mathbf{r}_{v,s} \leftarrow [\rho_{v,s}, \mu_{v,s}, \sigma_{v,s}, \delta_{v,s}]$ from valid timestamps
    $\mathbf{d}_{v,s} \leftarrow \text{LN}(\phi_{\text{feat}}(\bar{\mathbf{h}}_{v,s}) + \phi_{\text{stat}}(\mathbf{r}_{v,s}))$
    $\boldsymbol{\pi}_{v,s} \leftarrow \text{Softmax}_k(\mathcal{R}([\mathbf{d}_{v,s}; \mathbf{q}; \phi_I(\mathbf{g}^I); \mathbf{p}_v])/\tau)$
    Compute $\mathbf{e}_{v,s}^{\text{S}}$, $\mathbf{e}_{v,s}^{\text{M}}$, $\mathbf{e}_{v,s}^{\text{L}}$ and $\Delta\mathbf{h}_{v,s} \leftarrow \sum_k \pi_{v,s}^{(k)} \mathbf{e}_{v,s}^{(k)}$
  end for
  $\widetilde{\mathbf{h}}_{v,\ell}^T \leftarrow \mathbf{h}_{v,\ell}^T + \alpha\Delta\mathbf{h}_{v,s(\ell)}, \quad \alpha \leftarrow \alpha_{\max}\,\text{sigmoid}(a)$
  Pool $\widetilde{\mathbf{h}}_{v,\ell}^T$ and $\mathbf{f}_v \leftarrow \mathcal{P}_V(\cdot)$; map $\mathbf{f}_v$ to the fusion hidden size
end for
for each query τ and variable v **do**
  $(\boldsymbol{\gamma}_{v,\tau}, \boldsymbol{\beta}_{v,\tau}) \leftarrow g([\mathbf{u}_v; \mathbf{q}_\tau])$
  $\tilde{\mathbf{f}}_{v,\tau} \leftarrow (1 + \boldsymbol{\gamma}_{v,\tau}) \odot \mathbf{f}_v + \boldsymbol{\beta}_{v,\tau}, \quad \mathbf{z}_{v,\tau} \leftarrow [\tilde{\mathbf{f}}_{v,\tau}, \mathbf{q}_\tau]$
end for
if forecasting **then**
  $\hat{y}_{v,\tau} \leftarrow w^\top \mathbf{z}_{v,\tau}$
else
  $\bar{\mathbf{z}} \leftarrow \text{mean}_{v \in \mathcal{V}} \mathbf{z}_v; \;\; \hat{y} \leftarrow \text{Softmax}(W\bar{\mathbf{z}} + b)$
end if

Table 4: Forecasting dataset statistics.

	Human Activity	USHCN	PhysioNet	MIMIC-III
# Samples	5400	26736	12000	23457
# Variables	12	5	36	96
# Avg. Length	120	163	74	46
Missing Ratio	75%	77.9%	88.4%	96.7%

Human Activity (Kaluža et al., 2010). Wearable sensors on the ankles, belt, and chest of five subjects record 12 three-dimensional position channels at irregular times. The recordings are cut into 5,400 windows of 4,000 ms. We observe the first 1,000, 2,000, or 3,000 ms and forecast the remaining 3,000, 2,000, or 1,000 ms.

USHCN (Menne et al., 2015). The U.S. Historical Climatology Network stores five meteorological variables at weather stations. Following the standard IMTS split, we keep 1,114 stations over 1996–2000, which yields 26,736 samples. Each sample uses 24 months of history to predict the next 1, 6, or 12 months.

PhysioNet (Silva et al., 2012). This collection contains 12,000 ICU stays with 36 clinical variables measured irregularly in the first 48 hours after admission. We take the first 24, 36, or 12 hours as history and query values in the following 24, 12, or 36 hours.

MIMIC-III (Johnson et al., 2016). The processed forecasting subset has 23,457 stays and 96 variables from the first 48 hours of ICU care. The observation and forecast windows match those used for PhysioNet: 24→24 h, 36→12 h, and 12→36 h.

B.2 CLASSIFICATION

Table 5: Classification dataset statistics.

	P19	P12	PAM
# Samples	38804	11988	5333
# Variables	34	36	17
# Avg. Length	36	74	78
#Classes	2	2	8
Missing Ratio	94.9%	88.4%	65.0%

P19 (Reyna et al., 2019). This is the PhysioNet 2019 sepsis early-prediction cohort. After Raindrop-style filtering of extremely short and long series (Zhang et al., 2022), each stay has 34 time-varying measurements, a static vector (age, sex, admission delay, ICU type, and ICU length of stay), and a binary label for sepsis within the next 6 hours. The table reports 38,804 samples. Source: `https://physionet.org/content/challenge-2019/1.0.0/`.

P12 (Goldberger et al., 2000). Starting from the PhysioNet 2012 challenge, 12 stays identified by SeFT (Horn et al., 2020) are removed, leaving 11,988 patients. Each record covers the first 48 hours of ICU stay with 36 sensors (weight excluded) and a 9-dimensional static vector that includes age and sex. The binary label is ICU length of stay: at most three days versus more than three days. Source: `https://physionet.org/content/challenge-2012/1.0.0/`.

PAM (Reiss & Stricker, 2012). PAM is an 8-way activity task built from PAMAP2 by dropping one subject and rare activities, giving 5,333 windows of 600 steps and 17 channels. Sixty percent of the observations are then deleted at random to obtain an irregular series. Source: `https://archive.ics.uci.edu/static/public/231/pamap2+physical+activity+monitoring.zip/`.

C TASK-SPECIFIC DECODERS

Forecasting and classification share $\mathbf{z}_{v,\tau}$ from Sec. 4.3. For forecasting, τ is a future time in $\mathcal{Q}$; for classification, τ is the last valid timestamp, as in Sec. 4.1.

Forecasting reads out one value per variable and query,

$$\hat{y}_{v,\tau} = w^\top \mathbf{z}_{v,\tau}. \tag{13}$$

Classification has no future query. Let $\mathcal{V}$ be the set of variables with at least one valid observation. A masked mean $\bar{\mathbf{z}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{z}_v$ is mapped to class scores by a linear layer, $\hat{y} = \mathrm{Softmax}(W\bar{\mathbf{z}} + b)$.

D BASELINES

Forecasting compares against the fifteen models below. Classification uses the Raindrop/ISTS-PLM official-split protocol and is restricted to GRU-D, SeFT, mTAN, Raindrop, Warpformer, and Hi-Patch. All models are run from the authors' public code with the recommended hyperparameters, on the same splits as ScaleIMTS. Per-model settings are in Appendix F.1.

GRU-D (Che et al., 2018) (forecasting and classification) feeds missingness masks and time gaps into a GRU. Implementation: `https://github.com/Han-JD/GRU-D`.

SeFT (Horn et al., 2020) (forecasting and classification) treats observations as a set and pools them with multi-head attention. Implementation: `https://github.com/BorgwardtLab/Set_Functions_for_Time_Series`.

mTAN (Shukla & Marlin, 2021) (forecasting and classification) attends from a set of reference times onto irregular events to form a fixed-length trajectory. Implementation: `https://github.com/reml-lab/mTAN`.

CRU (Schirmer et al., 2022) (forecasting) uses continuous recurrent units with locally linear latent dynamics. Implementation: `https://github.com/boschresearch/Continuous-Recurrent-Units`.

PrimeNet (Chowdhury et al., 2023) (forecasting) pre-trains a time-aware Transformer on irregular inputs. Implementation: `https://github.com/ranakroychowdhury/PrimeNet`.

Raindrop (Zhang et al., 2022) (forecasting and classification) builds a sample-wise sensor graph and passes messages along time-varying edges. Implementation: `https://github.com/mims-harvard/Raindrop`.

Warpformer (Zhang et al., 2023) (forecasting and classification) aligns a series onto several warped resolutions and mixes them with dual attention. Implementation: `https://github.com/imJiawen/Warpformer`.

t-PatchGNN (Zhang et al., 2024) (forecasting) places irregular events into transformable time patches and runs a patch graph network. Implementation: `https://github.com/usail-hkust/t-PatchGNN`.

GraFITi (Yalavarthi et al., 2024) (forecasting) represents IMTS as a bipartite graph over timestamps and variables. Implementation: `https://github.com/yalavarthivk/GraFITi`.

APN (Liu et al., 2026b) (forecasting) aggregates events inside adaptive temporal patches with attention. Implementation: `https://github.com/decisionintelligence/APN`.

Hi-Patch (Luo et al., 2025) (forecasting and classification) stacks intra-patch and inter-patch graphs over a hierarchy of resolutions. Implementation: `https://github.com/qianlima-lab/Hi-Patch`.

HyperIMTS (Li et al., 2025) (forecasting) passes messages on a hypergraph that groups irregular observations. Implementation: `https://github.com/qianlima-lab/PyOmniTS`.

ReIMTS (Li et al., 2026) (forecasting) recursively splits the observation horizon and mixes tokens across those scales. We use the TimeCHEAT backbone. Implementation: `https://github.com/Ladbaby/PyOmniTS`.

KAFNet (Zhou et al., 2026) (forecasting) pools each variable with learnable Gaussian kernels in time, then attends across variables. We use the implementation integrated in our training pipeline. Implementation: `https://github.com/zhouziyu02/KAFNet`.

ISTS-PLM (Zhang et al., 2025) (forecasting) encodes IMTS as value-mask-time tokens for a pre-trained GPT temporal backbone and a BERT variable backbone. We use the authors' public implementation. Implementation: `https://github.com/usail-hkust/ISTS-PLM`.

E Performance Metrics

Forecasting is scored on queried values that are actually observed; classification is scored on sample-level labels.

Forecasting. Let N_{obs} be the number of test queries with $m_{v,\tau} = 1$. We report

$$\text{MSE} = \frac{1}{N_{\text{obs}}} \sum_{v,\tau} m_{v,\tau} (\hat{y}_{v,\tau} - y_{v,\tau})^2, \tag{14}$$

$$\mathrm{MAE} = \frac{1}{N_{\mathrm{obs}}} \sum_{v,\tau} m_{v,\tau} |\hat{y}_{v,\tau} - y_{v,\tau}|. \tag{15}$$

Classification. P12 and P19 are binary and class-imbalanced, so Table 2 reports AUROC and AUPRC from the positive-class probability. PAM is an eight-class task, so we report accuracy, precision, recall, and F1, with the last three averaged macro over classes. The classification panels of Fig. 4 use the relative AUPRC decrease over the clean test set. Ablation uses the same metrics as the main tables, from a single seed.

F More Implementation Details

F.1 Hyperparameters Settings

ScaleIMTS uses a frozen Timer encoder. The scale router uses $S = 4$ segments, three experts, temperature $\tau = 1$, and hidden width 256 for both the router and the expert MLPs. The residual gate is capped at $\alpha_{\max} = 0.3$, and the last linear layer of each expert is initialized to zero. The Event Patch Encoder uses hidden size 64, time encoding size 16, and node embedding size 16. Variable-level priors are mapped to dimension 128 before the affine rewrite. Input embeddings, variable prompts, the router, the experts, the Event Patch Encoder, the refinement MLP, and the task heads are trained; $\mathcal{P}_T$ and $\mathcal{P}_V$ are not. Baselines keep the authors' recommended hyperparameters and are not retuned.

Table 6 lists the training budget and patch length P used for ScaleIMTS. P is 100 on Human Activity, 20 on PAM, and 12 on the remaining datasets.

Table 6: ScaleIMTS training settings. P is the Event Patch Encoder width in time units.

Dataset	Batch	Learning rate	P
Human Activity	16	2×10^{-2}	100
USHCN	32	2×10^{-2}	12
PhysioNet	32	5×10^{-2}	12
MIMIC-III	8	1×10^{-3}	12
P12	8	1×10^{-3}	12
P19	8	1×10^{-3}	12
PAM	4	1×10^{-3}	20

F.2 Further Training Details

We train with Adam and stop when the validation loss does not improve for the patience. The checkpoint with the best validation loss is used for testing. Forecasting splits each dataset 6:2:2. Classification uses the official Raindrop/ISTS-PLM split files. Main study results are mean $\pm$ standard deviation over five random seeds $[2024, 2025, 2026, 2027, 2028]$. Ablation study, parameters study, and the local scale perturbation study use a single seed 2026. Perturbations are applied only at test time; training and validation remain clean.

F.3 Experimental Environment

All models are trained with the PyTorch library 2.10.0 and Python 3.11.15 on 2 GeForce RTX-4090D 24G GPUs.

G More Results of Experiments

G.1 Router Temperature

Figure 5 reports the complementary scan over the router temperature at $S = 4$. A small τ sharpens $\boldsymbol{\pi}_{v,s}$; a large τ mixes the three experts more uniformly. Only the clean test set is shown. The gray

band marks the default $\tau = 1$ used with $S = 4$ in the main experiments. Among the scanned temperatures, $\tau = 1$ attains the lowest MSE and MAE on USHCN and Human Activity, the highest AUPRC on P12, and the highest AUROC and AUPRC on P19.

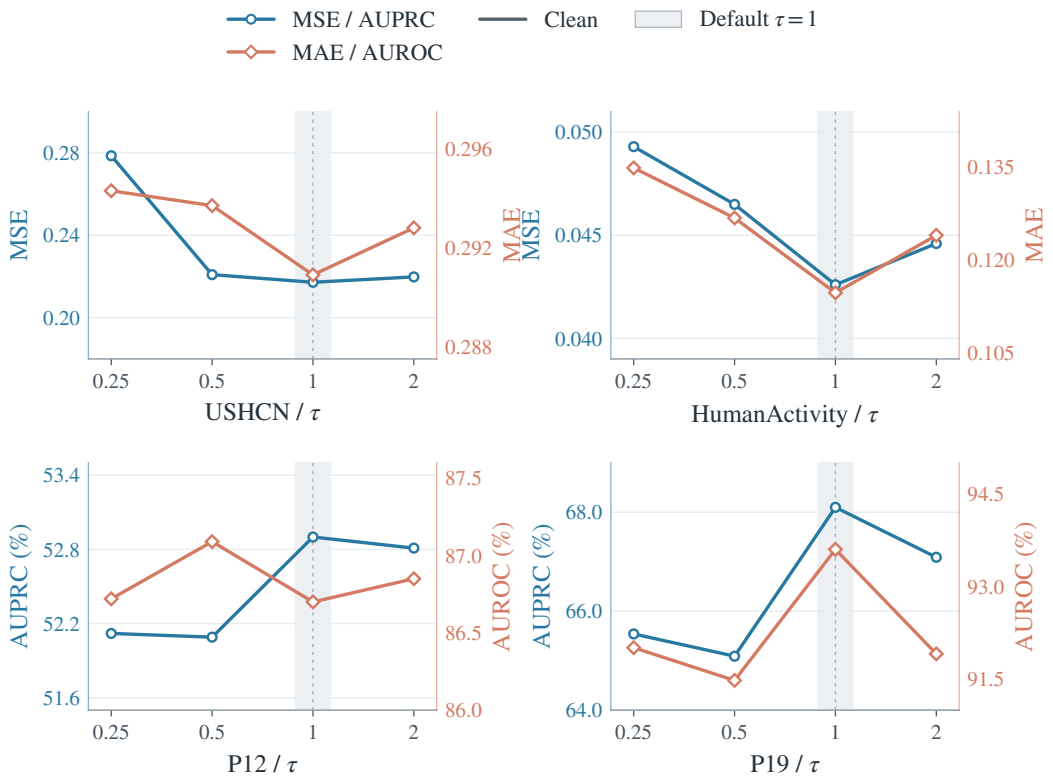


Figure 5: Sensitivity to the router temperature τ at $S = 4$ on the clean test set. Panel order and axis colors match Figure 3. The x-axis is $\log_2 \tau$. The gray band marks the default $\tau = 1$.

G.2 Perturbation Modes

The robustness evaluation uses a clean checkpoint and three test-time observation protocols. Random deletion thins events uniformly. local scale perturbation stretches intervals inside selected windows. Block missing removes contiguous spans. Figure 6 reports the relative MAE increase $(\mathrm{MAE}_{\mathrm{pert}}/\mathrm{MAE}_{\mathrm{clean}} - 1)$ on USHCN and Human Activity, and the relative AUPRC decrease on P12 and P19, for ScaleIMTS, w/o Local Scale, Hi-Patch, and KAFNet at removal rates 10%, 30%, and 50%.

ScaleIMTS remains the most stable of the four methods in every panel. It stays below Hi-Patch and KAFNet at all three rates, and the margin widens as more observations are removed. The largest gap appears under local scale perturbation at 50% removal, where the Human Activity MAE increase is 37% against 65%–75% for the two IMTS baselines, and the P19 AUPRC drop is 12% against 23%–27%. Disabling segment-wise routing raises these local scale entries, so the experts remain useful when short-scale windows are stretched. The same ranking holds under random deletion and block missing, which indicates that the scale-adapted prior is not specialized to a single missingness pattern.

G.3 Varying Observation and Forecast Horizons

The default window of each forecasting dataset is in Table 1. This section reports the remaining observation/forecast splits: Human Activity 2000→2000 and 1000→3000 ms, USHCN 24→6 and

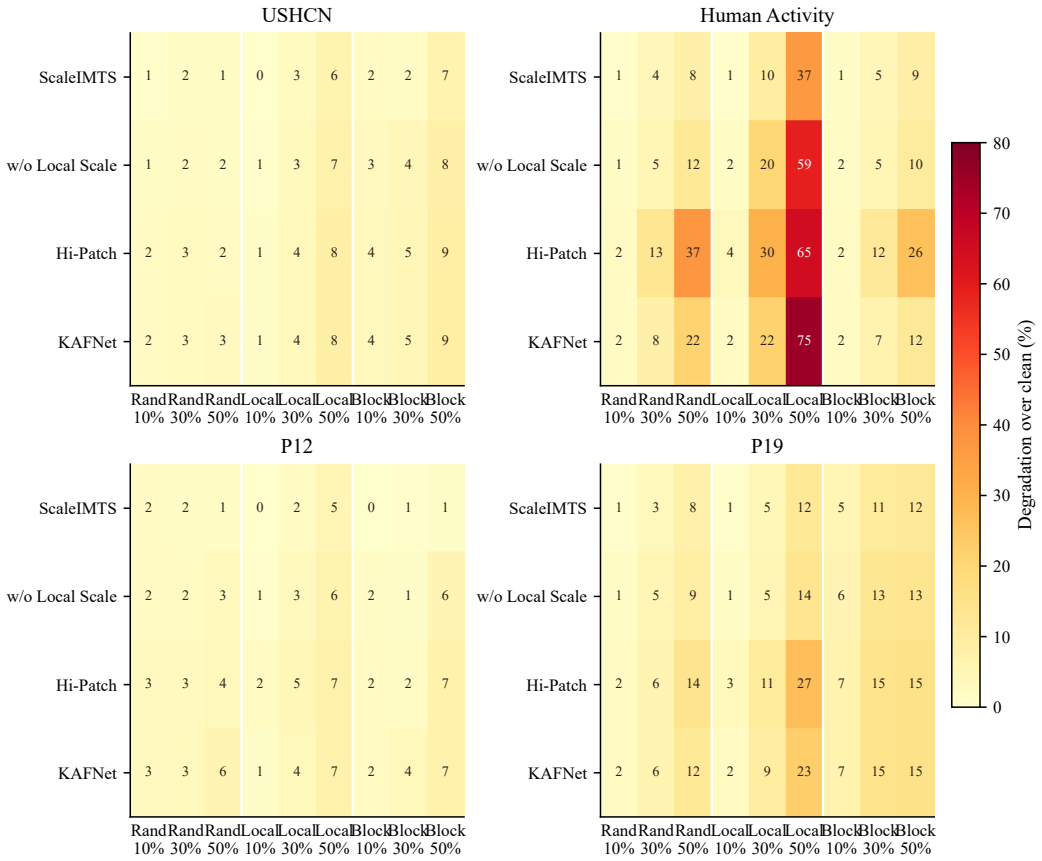


Figure 6: Relative degradation (%) under random, local scale, and block-missing perturbations at 10%, 30%, and 50% removal. Top: USHCN and Human Activity (MAE). Bottom: P12 and P19 (AUPRC).

24→12 months, and PhysioNet / MIMIC-III 36→12 and 12→36 h. Tables 7 and 8 use the same protocol as the main study.

ScaleIMTS remains the strongest overall method when the history is shortened or the forecast horizon is extended. On Human Activity it records the lowest MSE under both alternative windows and the lowest MAE under 2000→2000 ms. On USHCN it attains the lowest MSE and MAE for 24→6 months and the lowest MAE for 24→12 months. On PhysioNet it is best in both MSE and MAE for 36→12 h and 12→36 h. On MIMIC-III the longer 12→36 h window is likewise best in both metrics, while the shorter forecast stays competitive with the strongest graph and pretrained baselines. Relative to ISTS-PLM, the scale-adapted prior therefore transfers beyond the default horizon used in Table 1.

G.4 Resource Consumption

We compare wall-clock time per training epoch with the accuracy reported in the main tables (Fig. 7). Time is measured for one epoch on the same device and with the same batch size as the other methods in each panel. The vertical axis is MSE from Table 1 for forecasting, and AUPRC (P12, P19) or F1 (PAM) from Table 2 for classification, in percent. Although $\mathcal{P}_T$ and $\mathcal{P}_V$ are frozen, both still run in the forward pass, so the epoch time includes the pretrained encoders.

On Human Activity and USHCN, ScaleIMTS attains the lowest MSE at a mid-range epoch time: it is slower than APN, KAFNet, and GraFITi, and faster than CRU. On PhysioNet and MIMIC-III, where the number of variables and the missing ratio are larger, the epoch time of ScaleIMTS moves into the 10^1–10^2 s range. The MSE remains the best or second-best value in the panel, but

Table 7: Forecasting with alternative windows on Human Activity and USHCN. Results are MSE/MAE from a single run. Best in **bold**, second-best underlined among completed runs.

	Activity				USHCN			
	2000→2000 ms		1000→3000 ms		24→6 months		24→12 months	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
GRU-D	0.0868	0.1924	0.1106	0.2265	0.7741	0.4705	0.5598	0.3250
SeFT	0.1157	0.2340	0.1278	0.2505	0.6164	0.3554	0.5478	0.3418
mTAN	0.3531	0.4763	0.1664	0.3050	0.7890	0.4870	0.8158	0.4854
CRU	0.0973	0.2072	0.3031	0.4058	0.5646	0.3893	0.5408	0.3476
PrimeNet	0.3051	0.4075	0.3066	0.4086	0.8631	0.5155	0.8629	0.5190
Raindrop	0.3008	0.4017	0.3033	0.4055	0.8145	0.4815	0.6413	0.3989
Warpformer	0.1481	0.2744	0.1558	0.2876	0.8611	0.5101	0.8396	0.4780
t-PatchGNN	0.0874	0.1857	0.3537	0.4431	0.8626	0.5233	0.8024	0.5061
GraFITi	0.0827	0.1751	0.0979	**0.1885**	0.5815	0.3628	**0.5389**	0.3291
APN	0.0809	0.1710	0.0979	0.1905	0.6155	0.3544	0.5876	0.3651
Hi-Patch	0.0975	0.2064	0.1082	0.2197	0.7877	0.4893	0.5523	0.3250
HyperIMTS	0.0832	0.1728	0.1010	0.1980	0.7772	0.4894	0.7940	0.5056
ReIMTS	-	-	-	-	-	-	-	-
KAFNet	0.0833	0.1743	0.0986	0.1929	0.7245	0.4267	0.5424	0.3458
ISTS-PLM	0.0781	0.1720	0.0945	0.1939	0.7639	0.4657	0.8589	0.5126
ScaleIMTS	**0.0761**	**0.1691**	**0.0918**	0.1914	**0.5551**	**0.3253**	0.5414	**0.3215**

Table 8: Forecasting with alternative windows on PhysioNet and MIMIC-III. Results are MSE/MAE from a single run. Best in **bold**, second-best underlined among completed runs.

	PhysioNet				MIMIC-III			
	36→12 h		12→36 h		36→12 h		12→36 h	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
GRU-D	0.4875	0.5090	0.5957	0.5630	0.5377	0.4654	0.7308	0.5473
SeFT	0.7648	0.6646	0.7585	0.6531	0.6454	0.5230	0.7731	0.5665
mTAN	0.7618	0.6627	0.7595	0.6553	0.6632	0.5268	0.7772	0.5685
CRU	0.5500	0.5457	0.6025	0.5736	0.6541	0.5242	0.7811	0.5653
PrimeNet	0.7625	0.6637	0.7587	0.6540	0.9178	0.6350	0.9179	0.6338
Raindrop	0.7678	0.6698	0.7598	0.6552	0.9183	0.6348	0.9247	0.6378
Warpformer	0.7651	0.6664	0.7617	0.6616	0.5965	0.4975	0.7665	0.5717
t-PatchGNN	0.7678	0.6737	0.7617	0.6617	0.5206	0.4541	0.7615	0.5611
GraFITi	0.4274	0.4722	0.5514	0.5457	**0.4718**	0.4252	0.7218	0.5372
APN	0.3933	0.4342	0.5324	0.5390	0.5218	0.4507	0.7637	0.5626
Hi-Patch	0.7677	0.6735	0.5173	0.5223	0.5384	0.4511	0.7720	0.5707
HyperIMTS	-	-	-	-	0.5201	0.4616	0.7203	0.5362
ReIMTS	-	-	-	-	0.6266	0.4958	0.7874	0.5782
KAFNet	0.3964	0.4413	0.7617	0.6616	0.4986	0.4342	0.7587	0.5569
ISTS-PLM	0.7677	0.6735	0.5621	0.5571	0.4930	**0.4240**	0.7275	0.5429
ScaleIMTS	**0.3747**	**0.4231**	**0.4771**	**0.4917**	0.5014	0.4334	**0.6805**	**0.5134**

the method is no longer among the fastest. Relative to GraFITi on MIMIC-III, ScaleIMTS reduces MSE at a longer epoch; relative to CRU, Hi-Patch, and t-PatchGNN, it is both more accurate and cheaper per epoch.

Classification uses the same six baselines as Table 2. ScaleIMTS records the highest AUPRC on P12 and P19 and the highest F1 on PAM. Its epoch time lies between Warpformer and Hi-Patch on all three tasks, and is lower than GRU-D on PAM.

H Computational Complexity Analysis

The epoch-time trend in Fig. 7 is consistent with the cost of one forward pass, which we now make explicit. Let B be the batch size, L the padded history length, and V the number of variables, as in

Sec. 4.1. Let d_f and d_i be the hidden sizes of the foundation and irregular branches, H_T and H_V the numbers of temporal and variable backbone layers, S the number of local scale segments, R the number of scale experts, and n the number of time-aligned patches in Sec. 4.2. We analyze one forward pass of the full model.

The foundation branch tokenizes each variable independently and prepends a prompt, producing BV sequences of length $L+1$. The pretrained temporal backbone $\mathcal{P}_T$ applies self-attention to each sequence, which dominates this branch:

$$\mathcal{O}\big(H_T BV L^2 d_f\big). \tag{16}$$

The L non-prompt tokens are then split into S segments. A statistics encoder and a router assign each segment to R experts; every expert MLP is evaluated and the outputs are mixed by $\boldsymbol{\pi}_{v,s}$, at cost $\mathcal{O}(BVSRd_f)$. Pooled adapted tokens form a length-V sequence for the pretrained variable backbone $\mathcal{P}_V$,

$$\mathcal{O}\big(H_V BV^2 d_f\big). \tag{17}$$

Local scale routing is therefore linear in L and S, and does not add a second L^2 attention.

The irregular branch assigns observed events to n patches and aggregates each patch with a content-adaptive filter, which is linear in the padded event buffer. A Transformer then models each variable's patch sequence, $\mathcal{O}(BVn^2 d_i)$, and a dense graph on the V variables of every patch costs $\mathcal{O}(BnV^2 d_i)$. With $n \ll L$ and $d_i \ll d_f$, both terms are smaller than the temporal backbone.

Foundation refinement maps the variable-level prior, the irregular context, and the query into a shared space and applies a FiLM MLP, $\mathcal{O}(BVd_f)$. The task decoder is a linear readout of the same order.

Collecting the leading terms, one forward pass is

$$\mathcal{O}\big(H_T BV L^2 d_f + H_V BV^2 d_f + BVn^2 d_i + BnV^2 d_i\big). \tag{18}$$

The primary cost is the variable-wise temporal Transformer. The extra cost of ScaleIMTS relative to that encoder comes from local scale adaptation, patch-level irregular modeling, and FiLM refinement, all of lower order in L and d_f. Inference is this forward pass. Training uses the same forward computation; with frozen $\mathcal{P}_T$ and $\mathcal{P}_V$, the backbones still run in the forward pass but their weights are not updated.

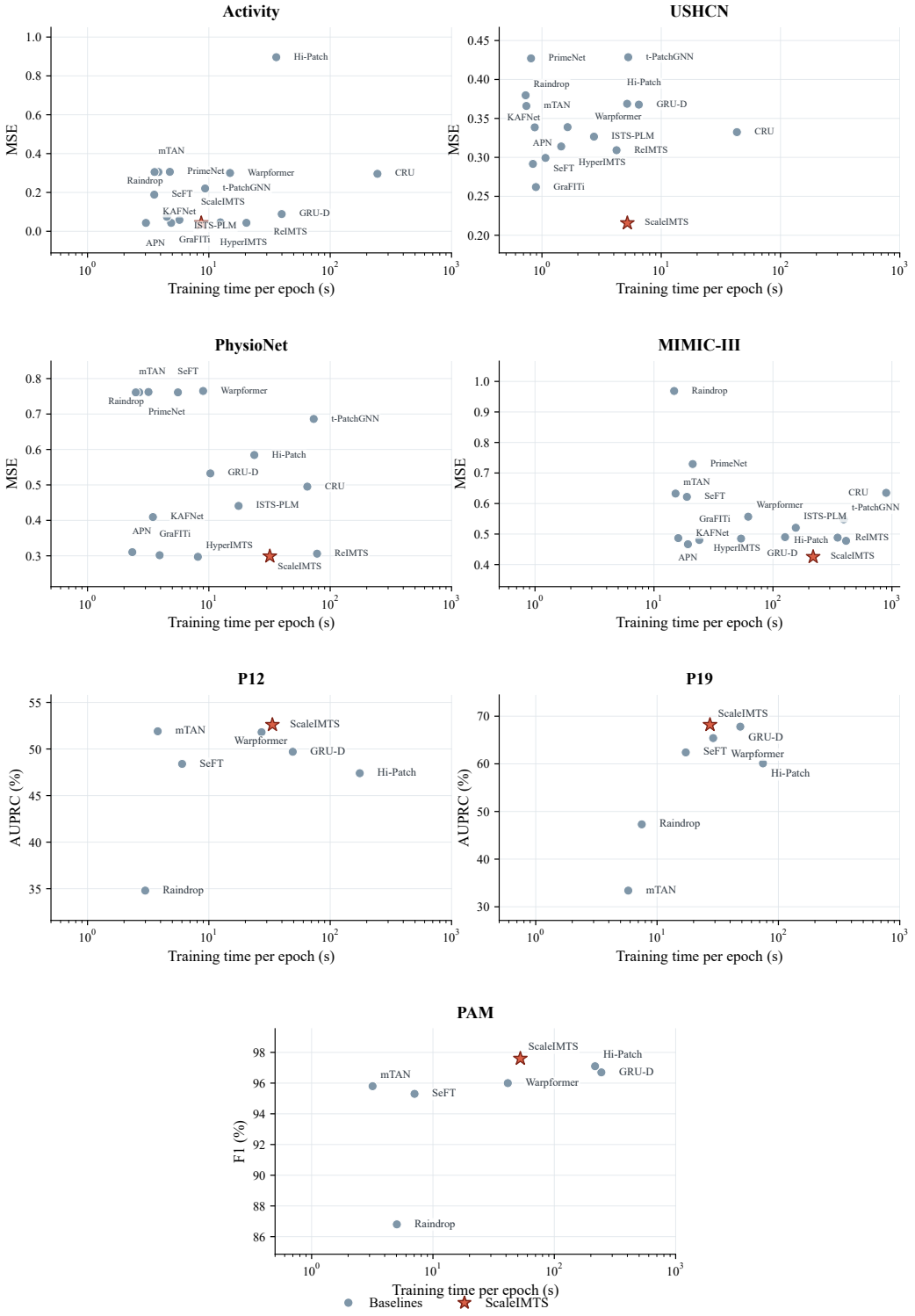


Figure 7: Accuracy versus training time per epoch. The x-axis is wall-clock time of one epoch (log scale). Forecasting panels use MSE; P12 and P19 use AUPRC and PAM uses F1, in percent.