HETEROGENEOUS PATTERN REPRESENTATION AND INTERACTION FOR GENERAL IRREGULAR MULTIVARIATE TIME SERIES ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series pattern machines (TSPMs) have been extensively studied for their ability to capture universal patterns and support diverse downstream tasks. However, existing TSPMs primarily target regularly sampled time series. In real-world scenarios, missing observations and differing sampling schedules often give rise to irregular multivariate time series (IMTS), characterized by uneven sampling intervals and asynchronous observations across variates. Developing TSPMs for IMTS is therefore more challenging because it requires capturing heterogeneous patterns, including temporal variations arising from intra-series irregularity and cross-variate dependencies arising from inter-series asynchrony. This calls for complementary representations of these patterns and effective interactions among them. To address these challenges, we propose **HeteroNet**, a TSPM designed for IMTS through **hetero**geneous pattern representation and interaction. It combines (1) Observation Encoding, which preserves the temporal context of individual observations, with (2) Sparse Pattern Tokenization, which explicitly represents recent states, changes over elapsed intervals, and variate histories. Then, (3) Pattern Interaction models dependencies among these heterogeneous patterns within and across asynchronous variates. Finally, (4) Kernel Fusion integrates the resulting representations with temporal kernel summaries of all observations. Extensive experiments across five IMTS analysis tasks show that HeteroNet achieves state-of-the-art performance, outperforming both general-purpose and task-specific baselines. Code is available at https://anonymous.4open.science/r/HeteroNet.

1 INTRODUCTION

General time series analysis aims to support diverse time series downstream tasks, including forecasting, classification, and anomaly detection (Wu et al., 2023; Luo & Wang, 2024; Xie et al., 2026), with applications spanning climate science, finance, and transportation (Huang et al., 2023; Wang et al., 2024; Shen et al., 2024; Ruan et al., 2026; Zhou et al., 2026b). Toward this goal, time series pattern machines (TSPMs) have attracted increasing attention for their ability to address these tasks within a unified architecture, reducing the need to develop separate task-specific models (Zhou et al., 2023; Liu et al., 2025b; Zhang et al., 2026a). Their central idea is to capture general temporal patterns and organize the underlying structures into representations that can be adapted to different downstream analysis tasks (Wu et al., 2023; Wang et al., 2025).

However, existing TSPMs are primarily designed for regularly sampled multivariate time series, where observations are evenly spaced in time and synchronized across variates. In real-world sensing systems, sensor failures, transmission errors, and sampling strategies designed to reduce acquisition costs often give rise to irregular multivariate time series (IMTS) (Liu et al., 2024; 2025a; 2026a; Zhou et al., 2026c; Liu et al., 2026b). These series exhibit uneven sampling intervals within individual variates and asynchronous observations across variates.

Although recent studies have addressed individual IMTS analysis tasks, such as classification (Liu et al., 2024; 2025a; Zhong et al., 2024), forecasting (Li et al., 2025; Luo et al., 2025), and generation (Zhang et al., 2026b), developing TSPMs for general IMTS analysis remains underexplored. The key challenge is to jointly capture temporal and cross-variate patterns arising from two defining characteristics of IMTS (as illustrated in Fig. 1): *intra-series irregularity* and *inter-series asynchrony* (Zhou et al., 2026c; Liu et al., 2026b). Within each variate, value changes over different

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.

intervals can indicate different rates of variation (long or short), making elapsed time essential for interpreting temporal behavior. Across variates, observations collected at different times and frequencies must still be related to uncover dependencies, without assuming temporal alignment.

To address these challenges, we propose **HeteroNet**, a time series pattern machine for general IMTS analysis through heterogeneous pattern representation and interaction. To model temporal variations arising from intra-series irregularity, **Observation Encoding** first embeds each value with its timestamp, elapsed interval, and variate identity, providing temporal context for individual observations. **Sparse Pattern Tokenization** then constructs observation, transition, and variate tokens to explicitly represent recent states, value changes together with their durations, and historical context for every variate within a compact sequence. To learn cross-variate dependencies arising from inter-series asynchrony, **Pattern Interaction** applies self-attention to these heterogeneous tokens, relating patterns within and across variates without requiring synchronized observations. The contextualized tokens are aggregated into sequence and variate representations. Finally, **Kernel Fusion** combines these representations with temporal kernel summaries from all observations, incorporating complementary local value and sampling information from the raw input. Task-specific output heads adapt the shared backbone architecture to different analysis tasks. Experiments on 15 datasets across classification, interpolation, extrapolation, temporal point process, and generation show that HeteroNet achieves state-of-the-art performance against 40 general-purpose and task-specific baselines.

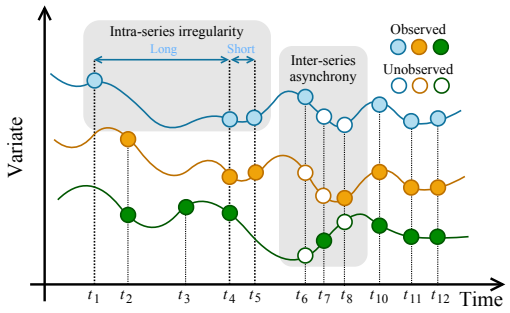


Figure 1: Illustration of IMTS. Long and short intervals between consecutive observations within a variate illustrate intra-series irregularity. At a given timestamp, some variates are observed while others are not, illustrating inter-series asynchrony.

Our contributions are summarized as follows:

- We introduce HeteroNet, a TSPM for general IMTS analysis that models temporal variations under irregular sampling and dependencies across asynchronously observed variates through heterogeneous pattern representation and interaction.
- HeteroNet captures a compact representation of observation, transition, and variate patterns through pattern interaction and kernel fusion. This unified backbone combines observed states, explicit temporal changes, and historical context.
- Experiments on 15 datasets spanning five IMTS analysis tasks show that HeteroNet marks a new state-of-the-art, outperforming both general-purpose and task-specific baselines.

2 Background and Problem Definitions

Previous evaluations of general IMTS methods have focused mainly on classification, interpolation, and extrapolation (Zhang et al., 2025; Liu et al., 2025a). To assess a broader range of capabilities, we also consider temporal point process modeling and generation. Temporal point process modeling describes the time and type of the next event, with applications in healthcare and online systems (Xue et al., 2024). Generation learns from irregularly observed training series to synthesize new multivariate trajectories, supporting simulation and data augmentation (Zhang et al., 2026b). These five tasks jointly assess sequence-level discrimination, value prediction, event dynamics, and generative modeling. Fig. 2 illustrates the tasks.

In this paper, we represent an irregular multivariate time series (IMTS) as $\mathcal{O} = \{(x_i, \tau_i, c_i)\}_{i=1}^{N_o}$, where N_o is the number of observations, $x_i \in \mathbb{R}$ is an observed value, τ_i is its timestamp, and $c_i \in \{1, \ldots, C\}$ identifies one of the C variates. A query (t, c) specifies a time t and variate c, and $\mathcal{Q} = \{(t_j, c_j)\}_{j=1}^{|\mathcal{Q}|}$ denotes a set of such queries. Specifically, the five tasks are defined as follows:

Classification. Given observations $\mathcal{O}$, the task is to predict a discrete class label $\widehat{y}$ for the entire series: $\mathcal{F}_{\text{cls}}(\mathcal{O}) \longrightarrow \widehat{y}$, where $\mathcal{F}_{\text{cls}}$ denotes the classification model to be learned.

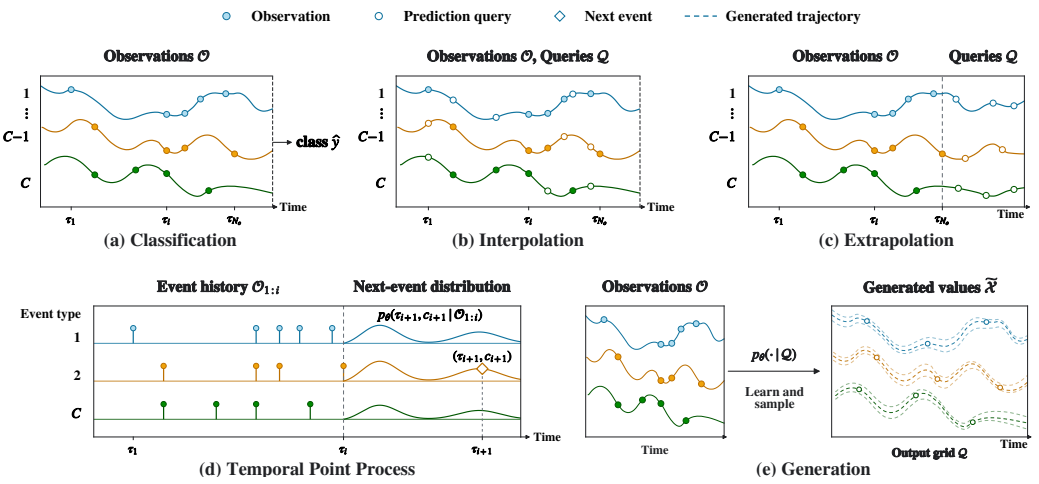


Figure 2: Five downstream tasks for IMTS analysis: (a) classification, (b) interpolation, (c) extrapolation, (d) temporal point process modeling, and (e) generation.

Interpolation. Given observations $\mathcal{O}$ and queries $\mathcal{Q}$ within the observed time window, the task is to estimate the corresponding unobserved values: $\mathcal{F}_{\text{int}}(\mathcal{O}, \mathcal{Q}) \longrightarrow \widehat{\mathcal{X}}$, where $\mathcal{F}_{\text{int}}$ denotes the value prediction model and $\widehat{\mathcal{X}} = \{\widehat{x}_j\}_{j=1}^{|\mathcal{Q}|}$ contains its predictions at the queried times and variates.

Extrapolation. Given observations $\mathcal{O}$ and queries $\mathcal{Q}$ after the last observed timestamp, the task is to predict future values using the mapping $\mathcal{F}_{\text{ext}}(\mathcal{O}, \mathcal{Q}) \longrightarrow \widehat{\mathcal{X}}$, with $t_j > \max_i \tau_i$ for every query.

Temporal Point Process. For marked event sequences, c_i denotes the event type and $x_i = 1$. Let $\mathcal{O}_{1:i}$ denote the first i events ordered by time. Given this history, the task is to learn the conditional distribution $p_\theta(\tau_{i+1}, c_{i+1} \mid \mathcal{O}_{1:i})$ of the next event's timestamp and type, where θ denotes learned model parameters. In next-event evaluation, both the timestamp and event type are predicted from the observed history, with type probabilities estimated over sampled next-event times.

Generation. Given a training collection of IMTS $\mathcal{D}_{\text{train}} = \{\mathcal{O}^{(n)}\}_{n=1}^{N}$, the task is to learn a generative model and sample new sequences on a prescribed output grid $\mathcal{Q}$: $\widetilde{\mathcal{X}} \sim p_\theta(\cdot \mid \mathcal{Q})$. Here, $\mathcal{Q}$ specifies output times and variates, and $\widetilde{\mathcal{X}} = \{\widetilde{x}_j\}_{j=1}^{|\mathcal{Q}|}$ contains the generated values.

A review of existing TSPMs and IMTS analysis can be found in Appendix A.

3 Methodology

Fig. 3 provides an overview of HeteroNet. First, **Observation Encoding** (Sec. 3.1) encodes the temporal context of individual observations under irregular sampling. **Sparse Pattern Tokenization** (Sec. 3.2) then constructs complementary tokens representing recent states, changes over elapsed intervals, and variate histories. **Pattern Interaction** (Sec. 3.3) models dependencies among these tokens within and across asynchronously observed variates. The resulting tokens are aggregated into sequence-level and variate-level representations. **Kernel Fusion** (Sec. 3.4) combines these representations with temporal kernel summaries computed from all observations in a parallel branch. Finally, task-specific heads map the learned representations to task outputs.

3.1 Observation Encoding

In IMTS, similar observed values can occur at different temporal positions or follow substantially different sampling gaps. To preserve these intra-series distinctions and model these temporal variations, we encode each value together with its timestamp, elapsed interval, and variate identity. Let $\tau_{\min}$ and $\tau_{\max}$ be the earliest and latest timestamps in $\mathcal{O}$. We first normalize time as $\bar{\tau}_i = (\tau_i - \tau_{\min})/(\tau_{\max} - \tau_{\min})$, where i indexes individual observations in $\mathcal{O}$. Let δ_i be the normalized time since the preceding observation of the same variate, with zero for its first observation. The temporal embedding $\mathbf{E}_t$ and the corresponding observation representation $\mathbf{e}_i \in \mathbb{R}^d$ are:

$$\begin{aligned} \mathbf{E}_t(\bar{\tau}_i) &= A_t(\text{Concat}(\bar{\tau}_i, \sin(\boldsymbol{\omega}\bar{\tau}_i), \cos(\boldsymbol{\omega}\bar{\tau}_i))) \,, \\ \mathbf{e}_i &= \text{LN}\big(\text{FFN}\big(A_x(x_i) + \mathbf{E}_t(\bar{\tau}_i) + A_\delta(\delta_i) + \mathbf{u}_{c_i}^{\text{enc}}\big)\big) \,. \end{aligned} \tag{1}$$

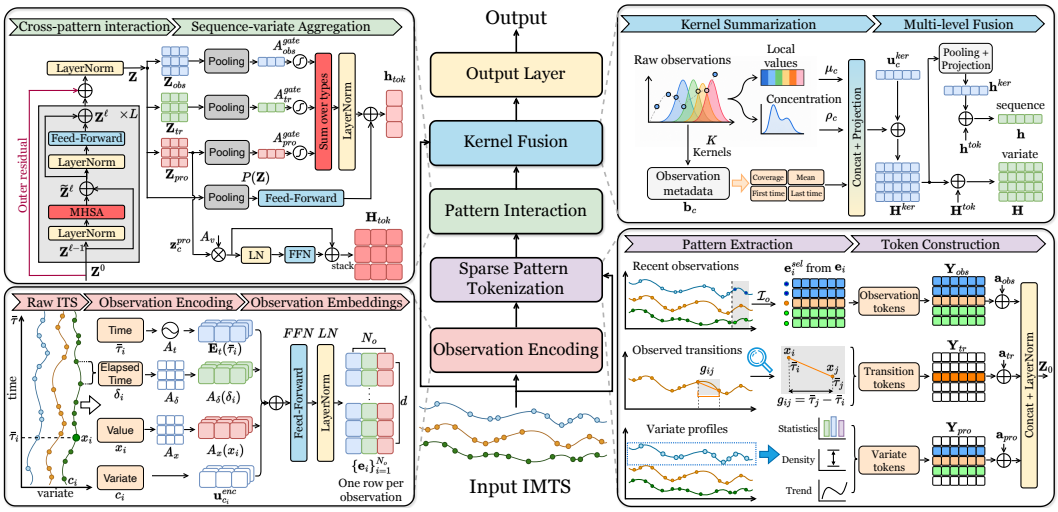


Figure 3: Overview of HeteroNet. Observation Encoding embeds each observation with its temporal and variate context. Sparse Pattern Tokenization constructs observation, transition, and profile tokens to represent recent states, temporal changes, and variate histories. Pattern Interaction models dependencies among these tokens and produces sequence-level and variate-level representations. Kernel Fusion combines these representations with temporal kernel summaries computed from all observations. A task-specific output layer maps the fused representations to downstream outputs.

Here, $\boldsymbol{\omega}$ contains fixed frequencies, and $\mathrm{Concat}(\cdot)$ denotes feature concatenation. A_t, A_x, A_δ are learnable projections into $\mathbb{R}^d$, where d denotes representation dimension. $\mathbf{u}_c^{\mathrm{enc}}$ is a learnable variate embedding, $\mathrm{FFN}(\cdot)$ is a learnable feed-forward network, and $\mathrm{LN}(\cdot)$ denotes layer normalization. The output $\mathbf{e}_i$ provides a temporally informed foundation for learning patterns from irregular observations.

3.2 Sparse Pattern Tokenization

Observation Encoding provides temporal context for individual observations, while changes between observations and overall variate behavior remain implicit. Moreover, retaining only recent observations can omit sparsely observed variates. To explicitly capture temporal changes while preserving historical context, we construct three complementary token types representing recent states, changes over elapsed intervals, and variate histories.

Observation tokens. To preserve recent states, we select up to B_o observations with the largest timestamps across all variates. Their original observation indices form $\mathcal{I}_o$, identifying the encoder outputs to retain. The selected embeddings $\{\mathbf{e}_i : i \in \mathcal{I}_o\}$ form the rows of $\mathbf{Y}_{\mathrm{obs}} \in \mathbb{R}^{|\mathcal{I}_o| \times d}$.

Transition tokens. To associate value changes with their durations, we consider observed (i, j) pairs from the same variate at adjacent original time-ordered input positions, with j following i. We retain in the pair index set $\mathcal{I}_g$ up to B_g pairs with the largest normalized intervals $g_{ij} = \bar{\tau}_j - \bar{\tau}_i$. A learnable projection encodes each pair's duration, endpoint times endpoint values, and value difference $x_j - x_i$, directly from raw observation values and normalized timestamps. Adding a learnable variate embedding produces $\mathbf{Y}_{\mathrm{tr}} \in \mathbb{R}^{|\mathcal{I}_g| \times d}$, explicitly representing how much values change over each selected interval.

Variate tokens. To retain historical context beyond selected observations and transitions, we construct one profile from the full observed history of each variate. *Statistics* comprise the mean, standard deviation, minimum, and maximum of the observed values x of each variate. *Density* captures the observation count and the mean and maximum time intervals between observations at adjacent input positions within each variate. *Trend* records endpoint values, their difference, and change per unit elapsed time. A learnable projection of these concatenated features, followed by addition of a variate embedding, produces $\mathbf{Y}_{\mathrm{pro}} \in \mathbb{R}^{C \times d}$.

Let $\mathcal{R} = \{\mathrm{obs}, \mathrm{tr}, \mathrm{pro}\}$ denote the observation, transition, and variate token types. For each type r, $\mathbf{Y}_r \in \mathbb{R}^{N_r \times d}$ contains its tokens as rows, where $N_{\mathrm{obs}} = |\mathcal{I}_o|$, $N_{\mathrm{tr}} = |\mathcal{I}_g|$, and $N_{\mathrm{pro}} = C$. We add a

type embedding $\mathbf{a}_r$ to each row of $\mathbf{Y}_r$ to distinguish the roles of states, transitions, and histories:

$$\mathbf{Z}^0 = \mathrm{LN}(\mathrm{Concat}\left(\{\mathbf{Y}_r + \mathbf{a}_r\}_{r\in\mathcal{R}}\right)) \in \mathbb{R}^{M\times d}. \tag{2}$$

Here, $\mathrm{Concat}(\cdot)$ stacks rows in observation, transition, and variate order, and $M = |\mathcal{I}_o| + |\mathcal{I}_g| + C \leq B_o + B_g + C$ is the number of valid tokens. The resulting $\mathbf{Z}^0$ organizes recent states, interval-level changes, and variate histories into a common representation space. Their complementary roles provide a compact basis for learning temporal and cross-variate dependencies through pattern interaction.

3.3 Pattern Interaction

In IMTS, related behavior can occur at different times across variates, making cross-variate dependencies difficult to model. We jointly contextualize the three token types through self-attention, allowing recent states, changes, and histories to inform one another within and across variates without synchronizing observations. Starting from $\mathbf{Z}^0$, we apply L attention-based encoder layers. Layer $\ell = 1, \ldots, L$ updates its input $\mathbf{Z}^{\ell-1}$ as:

$$\widetilde{\mathbf{Z}}^{\ell} = \mathbf{Z}^{\ell-1} + \mathrm{MHSA}_{\ell}\big(\mathrm{LN}(\mathbf{Z}^{\ell-1})\big), \quad \mathbf{Z}^{\ell} = \widetilde{\mathbf{Z}}^{\ell} + F_{\ell}\Big(\mathrm{LN}(\widetilde{\mathbf{Z}}^{\ell})\Big). \tag{3}$$

Here, $\mathrm{MHSA}_{\ell}(\cdot)$ denotes multi-head self-attention, $F_{\ell}(\cdot)$ is a feed-forward network applied to each row, and $\mathbf{Z}^{\ell}$ is the output of the ℓ-th layer. An outer residual gives the contextualized pattern tokens $\mathbf{Z} = \mathrm{LN}(\mathbf{Z}^0 + \mathbf{Z}^L) \in \mathbb{R}^{M\times d}$. Thus, $\mathbf{Z}$ encodes dependencies among observed states, temporal transitions, and variate histories within and across asynchronously observed variates. To retain both global context specific to each variate, we aggregate $\mathbf{Z}$ into sequence and variate representations.

Sequence representation. We combine type-specific summaries with a pooled summary over all tokens. Let $\mathbf{Z}_r$ be the submatrix containing the rows of $\mathbf{Z}$ belonging to type r. The type summary $\mathbf{p}_r \in \mathbb{R}^d$ and sequence representation $\mathbf{h}^{\mathrm{tok}} \in \mathbb{R}^d$ are:

$$\begin{aligned} \mathbf{p}_r &= \mathrm{LN}(\mathrm{GELU}(A_r(P(\mathbf{Z}_r)))), \\ \mathbf{h}^{\mathrm{tok}} &= \mathrm{FFN}(P(\mathbf{Z})) + \mathrm{LN}\left(\sum_{r\in\mathcal{R}} \mathbf{p}_r \odot \mathrm{Sigmoid}(A_r^{\mathrm{gate}}(\mathbf{p}_r))\right), \end{aligned} \tag{4}$$

where $P(\mathbf{U}) = \mathrm{Concat}(\mathrm{Mean}(\mathbf{U}), \mathrm{Max}(\mathbf{U}))$, with $\mathrm{Mean}(\mathbf{U})$ and $\mathrm{Max}(\mathbf{U})$ denoting the column-wise mean and maximum vectors, respectively. A_r and A_r^{gate} are learnable summary and gating projections. $\odot$ denotes elementwise multiplication. The gates weight each type's contribution by feature, while the first term captures their combined information.

Variate representations. The contextualized variate token $\mathbf{z}_c^{\mathrm{pro}}$, the row of $\mathbf{Z}_{\mathrm{pro}}$ for variate c, combines its profile with information from other patterns. We refine it as:

$$\mathbf{h}_c^{\mathrm{tok}} = A_v(\mathbf{z}_c^{\mathrm{pro}}) + \mathrm{FFN}(\mathrm{LN}\left(A_v(\mathbf{z}_c^{\mathrm{pro}})\right)). \tag{5}$$

The learnable projection A_v and $\mathrm{FFN}(\cdot)$ are shared across variates. Stacking these representations as rows gives $\mathbf{H}^{\mathrm{tok}} \in \mathbb{R}^{C\times d}$. Together, $\mathbf{h}^{\mathrm{tok}}$ and $\mathbf{H}^{\mathrm{tok}}$ summarize temporal and cross-variate patterns at the sequence and variate levels, preserving dependencies learned under inter-series asynchrony.

3.4 Kernel Fusion

Pattern Interaction produces sequence and variate representations that capture dependencies among heterogeneous patterns. To complement these compact representations with local value and sampling information throughout the input window, Kernel Fusion computes temporal kernel summaries from all observations and integrates them with the token representations at the sequence and variate levels.

We use K fixed centers ξ_k uniformly spaced on $[0, 1]$ and learnable bandwidths $\sigma_k > 0$. Kernel k assigns normalized time t the weight $\omega_k(t) = \exp(-(t-\xi_k)^2/(2\sigma_k^2))$. For variate c with $n_c > 0$ observations, we compute:

$$\mu_{ck} = \frac{\sum_{i:c_i=c} \omega_k(\bar{\tau}_i) x_i}{\sum_{i:c_i=c} \omega_k(\bar{\tau}_i)}, \qquad \rho_{ck} = \frac{1}{n_c} \sum_{i:c_i=c} \omega_k(\bar{\tau}_i). \tag{6}$$

The weighted mean μ_{ck} summarizes values near ξ_k, while the average kernel weight ρ_{ck} describes the concentration of observation times around it. Let $\boldsymbol{\mu}_c = (\mu_{c1}, \ldots, \mu_{cK})$ and $\boldsymbol{\rho}_c = (\rho_{c1}, \ldots, \rho_{cK})$

collect these summaries. Let L_{in} denote the padded input length before tokenization. Metadata $\mathbf{b}_c$ contains the normalized observation count n_c/L_{in}, the mean observed value, and the first and last normalized observation times of variate c. The kernel representations are:

$$\mathbf{h}_c^{\text{ker}} = \text{FFN}(\text{Concat}(\boldsymbol{\mu}_c, \boldsymbol{\rho}_c, \mathbf{b}_c)) + \mathbf{u}_c^{\text{ker}}, \quad \mathbf{h}^{\text{ker}} = \text{FFN}(P(\mathbf{H}^{\text{ker}})). \tag{7}$$

$\mathbf{u}_c^{\text{ker}}$ is a separate learnable variate embedding. $\mathbf{H}^{\text{ker}} \in \mathbb{R}^{C\times d}$ stacks $\mathbf{h}_c^{\text{ker}}$ as rows, and $\mathbf{h}^{\text{ker}} \in \mathbb{R}^d$ summarizes the sequence. We add the two branches at matching levels:

$$\mathbf{H} = \mathbf{H}^{\text{tok}} + \mathbf{H}^{\text{ker}}, \quad \mathbf{h} = \mathbf{h}^{\text{tok}} + \mathbf{h}^{\text{ker}}. \tag{8}$$

$\mathbf{H} \in \mathbb{R}^{C\times d}$ and $\mathbf{h} \in \mathbb{R}^d$ provide variate and sequence representations, respectively, combining temporal and cross-variate dependencies with local value and sampling information from the input.

To support different IMTS tasks, task-specific heads use $\mathbf{Z}$, $\mathbf{H}$, and $\mathbf{h}$ as required, with a separate model trained for each task. Output-head architectures are provided in Appendix B.

4 Experiments

We evaluate HeteroNet as a general IMTS pattern machine across five mainstream tasks: classification, interpolation, extrapolation, temporal point process modeling, and generation. We follow established evaluation protocols for classification, interpolation, and extrapolation (Zhang et al., 2025), temporal point process modeling (Xue et al., 2024), and generation (Zhang et al., 2026b), covering 15 datasets and 40 baselines in total. Detailed descriptions of the datasets, baselines, evaluation metrics, and implementation settings are provided in Appendices C.1, C.2, C.3, and C.4, respectively.

4.1 Main Results

Fig. 4 compares performance across the five tasks, using one representative metric per task averaged over the corresponding datasets. HeteroNet achieves the best average performance on all five tasks among the compared models (Zhang et al., 2024a; 2025; Liu et al., 2025a; Yalavarthi et al., 2024; Zhou et al., 2026c; Liu et al., 2026b; Qiu et al., 2026). The main text presents comparisons with the strongest baselines for each task. Additional results in Appendix E show that HeteroNet maintains strong performance across the five tasks against an expanded set of general-purpose and task-specific baselines.

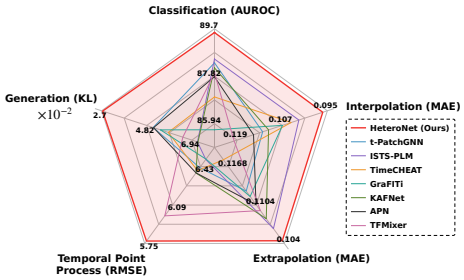


Figure 4: Overall performance comparison.

4.1.1 Classification

Setups. Classification supports applications such as clinical assessment, human activity recognition, and fault diagnosis (Zhang et al., 2022; 2023). Following (Zhang et al., 2025), we evaluate HeteroNet on three datasets spanning clinical and activity recognition settings: PhysioNet 2012 (P12) (Silva et al., 2012), PhysioNet 2019 (P19) (Reyna et al., 2019), and PAM (Reiss & Stricker, 2012). Following established evaluation protocols (Zhang et al., 2025; Liu et al., 2025a), we report AUROC and AUPRC on P12 and P19, and accuracy, precision, recall, and F1 score on PAM.

Table 1: Results on classification (mean ± std; higher is better). Best and second-best values are highlighted in **red** and blue.

Method	P12		P19		PAM			
	AUROC	AUPRC	AUROC	AUPRC	Accuracy	Precision	Recall	F1
t-PatchGNN	86.1 ± 1.1	56.4 ± 0.9	90.6 ± 1.3	57.3 ± 2.9	96.1 ± 0.4	96.4 ± 0.7	96.9 ± 0.3	96.4 ± 1.3
ISTS-PLM	**87.6** ± 1.4	**57.6** ± 3.3	89.4 ± 2.2	56.9 ± 5.0	96.3 ± 0.5	96.9 ± 1.0	96.8 ± 0.4	96.8 ± 0.7
TimeCHEAT	84.5 ± 0.7	48.2 ± 1.9	89.5 ± 1.9	56.1 ± 4.6	96.5 ± 0.6	97.1 ± 0.5	96.9 ± 0.6	97.0 ± 0.5
KAFNet	86.2 ± 0.4	52.1 ± 0.8	90.1 ± 0.3	59.3 ± 0.6	95.5 ± 0.6	97.1 ± 0.6	96.1 ± 0.4	95.5 ± 0.8
APN	85.7 ± 0.5	53.3 ± 0.9	89.9 ± 0.2	58.9 ± 0.7	95.6 ± 0.5	96.8 ± 0.5	95.9 ± 0.4	95.3 ± 0.4
TFMixer	86.3 ± 0.6	54.3 ± 0.7	89.6 ± 0.5	58.7 ± 0.8	95.8 ± 0.3	96.8 ± 0.6	96.1 ± 0.5	95.5 ± 0.5
HeteroNet	87.3 ± 0.3	54.1 ± 1.1	**91.8** ± 0.5	**62.2** ± 0.5	**97.0** ± 0.1	**97.8** ± 0.1	**97.3** ± 0.1	**97.5** ± 0.1

Results. Tab. 1 shows the clearest classification gains on P19 and PAM, while the P12 result depends on the metric. This pattern is consistent with representing recent states, changes over elapsed intervals, and variate histories together, although their utility varies across classification targets. Additional results in Appendix E.1 show that HeteroNet generally outperforms other baselines.

4.1.2 INTERPOLATION

Setups. Interpolation supports applications such as recovering missing clinical measurements and reconstructing incomplete sensor records (Liu et al., 2025a). Following (Zhang et al., 2025), we evaluate HeteroNet on four datasets: PhysioNet (Silva et al., 2012), MIMIC (Johnson et al., 2016), Human Activity (Vidulin et al., 2010), and USHCN (Menne et al., 2016). The task predicts unobserved values within the observed time window. We report mean squared error (MSE) and mean absolute error (MAE) following existing works (Zhang et al., 2025; Liu et al., 2025a).

Table 2: Results on interpolation measured by MSE and MAE (mean ± std, lower is better). Best and second-best values are highlighted in **red** and blue, respectively.

Method	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}
t-PatchGNN	4.75 ± 0.03	3.59 ± 0.03	1.55 ± 0.08	6.25 ± 0.21	1.95 ± 0.12	2.54 ± 0.14	5.63 ± 0.04	3.25 ± 0.03
ISTS-PLM	4.55 ± 0.08	3.37 ± 0.02	1.47 ± 0.01	5.94 ± 0.01	1.93 ± 0.01	2.52 ± 0.01	5.45 ± 0.08	2.92 ± 0.06
TimeCHEAT	5.18 ± 0.08	3.55 ± 0.06	1.42 ± 0.08	5.91 ± 0.05	2.12 ± 0.11	2.89 ± 0.04	5.46 ± 0.07	2.95 ± 0.06
KAFNet	6.52 ± 0.03	4.26 ± 0.05	1.58 ± 0.08	6.12 ± 0.11	2.93 ± 0.03	3.37 ± 0.08	7.68 ± 0.06	3.05 ± 0.04
APN	6.12 ± 0.04	4.04 ± 0.08	1.66 ± 0.05	6.02 ± 0.03	1.93 ± 0.05	2.89 ± 0.06	7.78 ± 0.03	3.29 ± 0.02
TFMixer	5.53 ± 0.04	4.04 ± 0.06	1.88 ± 0.07	5.92 ± 0.06	2.24 ± 0.05	3.05 ± 0.02	7.06 ± 0.08	3.49 ± 0.05
HeteroNet	**4.31** ± 0.03	**3.13** ± 0.03	**1.12** ± 0.04	**5.38** ± 0.11	**1.76** ± 0.01	**2.28** ± 0.02	**5.06** ± 0.01	**2.77** ± 0.01

Results. Tab. 2 shows consistent gains in reconstructing values within the observed time window. The selected temporal patterns and summaries of all observations provide complementary context for within-window queries. Additional results in Appendix E.2 show that HeteroNet retains the lowest MSE and MAE on all four datasets against all the baselines.

4.1.3 EXTRAPOLATION

Setups. Extrapolation supports applications such as forecasting clinical measurements, human motion, and climate variables from irregularly sampled histories (Zhang et al., 2024a; 2025; 2024b). Following (Zhang et al., 2025), we evaluate HeteroNet on PhysioNet (Silva et al., 2012), MIMIC (Johnson et al., 2016), Human Activity (Vidulin et al., 2010), and USHCN (Menne et al., 2016), placing all prediction queries beyond the observed time window.

Table 3: Results on extrapolation measured by MSE and MAE (mean ± std, lower is better). Best and second-best values are highlighted in **red** and blue, respectively.

Method	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}
t-PatchGNN	4.98 ± 0.08	3.72 ± 0.03	1.69 ± 0.03	7.22 ± 0.09	2.66 ± 0.03	3.15 ± 0.02	5.00 ± 0.04	3.10 ± 0.04
ISTS-PLM	4.92 ± 0.05	3.65 ± 0.04	1.64 ± 0.02	7.02 ± 0.14	2.58 ± 0.03	3.12 ± 0.04	5.06 ± 0.03	2.88 ± 0.02
TimeCHEAT	5.07 ± 0.02	3.90 ± 0.01	1.70 ± 0.02	7.43 ± 0.06	4.05 ± 0.06	4.65 ± 0.03	**4.45** ± 0.03	3.12 ± 0.05
KAFNet	5.90 ± 0.02	3.54 ± 0.01	**1.59** ± 0.02	**6.84** ± 0.06	2.55 ± 0.05	3.07 ± 0.05	5.02 ± 0.04	2.98 ± 0.02
APN	5.04 ± 0.02	3.63 ± 0.04	1.85 ± 0.02	7.01 ± 0.04	2.63 ± 0.06	3.15 ± 0.04	5.13 ± 0.03	3.04 ± 0.04
TFMixer	5.01 ± 0.03	3.62 ± 0.05	1.88 ± 0.04	7.03 ± 0.06	2.60 ± 0.04	3.14 ± 0.02	5.11 ± 0.01	3.00 ± 0.06
HeteroNet	**4.72** ± 0.02	**3.46** ± 0.02	1.68 ± 0.06	6.98 ± 0.08	**2.53** ± 0.01	**3.04** ± 0.01	5.04 ± 0.01	**2.83** ± 0.02

Results. Tab. 3 shows that HeteroNet's advantage is more consistent on PhysioNet and Human Activity than on MIMIC and USHCN. Because queries lie beyond the observed window, this task probes how well the learned historical representations support future prediction. Results in Appendix E.3 show HeteroNet largely outperforms other baselines against the expanded baseline set.

4.1.4 TEMPORAL POINT PROCESS

Setups. Temporal point process (TPP) modeling supports applications such as predicting online interactions and transportation events (Zhou et al., 2026a). Following (Xue et al., 2024), we evaluate HeteroNet on four datasets: Amazon (Ni et al., 2019), Retweet (Zhou et al., 2013), Taxi (Whong, 2014), and Taobao (Xue et al., 2022). TPP is evaluated by root mean squared error (RMSE) for next-event time prediction and error rate (Err) for next-event type prediction.

Results. Tab. 4 shows stronger results for event-time prediction and next-event-type prediction. Observation and transition tokens encode recent events and elapsed intervals, while interactions across event types provide information for predicting both the next event's time and type. Additional

Table 4: Results on TPP measured by RMSE and Err (mean ± std, lower is better). Best and second-best values are highlighted in **red** and blue, respectively. "–" indicates unavailable event-type results for FullyNN (Omi et al., 2019).

Method	Amazon		Retweet		Taxi		Taobao	
	RMSE	Err(%)	RMSE	Err(%)	RMSE	Err(%)	RMSE	Err(%)
NHP	0.621±0.005	67.1±0.60	21.90±0.184	40.0±0.20	0.369±0.003	**8.50**±0.05	0.531±0.005	54.2±0.60
THP	0.621±0.003	66.1±0.70	22.01±0.188	41.5±0.30	0.370±0.003	8.68±0.06	0.531±0.003	53.6±0.40
AttNHP	0.621±0.005	**65.3**±0.60	22.19±0.180	40.1±0.30	0.371±0.003	8.71±0.04	**0.529**±0.005	53.7±0.10
ODETPP	0.620±0.006	65.8±0.80	22.48±0.175	43.2±0.40	0.371±0.003	10.54±0.08	0.533±0.005	55.4±0.70
FullyNN	0.615±0.005	–	21.92±0.159	–	0.373±0.003	–	**0.529**±0.005	–
IFTPP	0.618±0.005	67.5±0.70	22.18±0.204	**39.7**±0.30	0.377±0.003	8.56±0.60	0.531±0.005	55.4±0.40
HeteroNet	**0.514**±0.004	66.7±0.26	**21.65**±0.11	40.6±0.20	**0.368**±0.002	10.6±0.39	0.552±0.008	**41.9**±0.33

results in Appendix E.4 show that HeteroNet retains the lowest event-time RMSE on Amazon, Retweet and Taxi, and the lowest event-type error on Taobao.

4.1.5 GENERATION

Setups. Generation supports applications such as synthesizing time series for data augmentation and simulation (Zhang et al., 2026b; Deng et al., 2025; Shen et al., 2026). Following (Zhang et al., 2026b), we evaluate HeteroNet on Sines (Yoon et al., 2019; Jeon et al., 2022), Stocks (Yoon et al., 2019; Jeon et al., 2022), Energy (Candanedo, 2017), and MuJoCo (Todorov et al., 2012) against six dedicated generation baselines shown in the main table. We report discriminative score (DS), marginal distribution difference (MDD), and Kullback–Leibler divergence (KL), averaging each metric over missing rates of 30%, 50%, and 70% and sequence lengths of 12, 24, and 36.

Table 5: Results on generation, averaged over all missing rates and sequence lengths (point estimates; lower is better). DS, MDD, and KL are reported in units of 10^{-2}, 10^{-1}, and 10^{-2}, respectively. Best and second-best values are highlighted in **red** and blue.

Method	Sines			Stocks			Energy			MuJoCo		
	DS	MDD	KL	DS	MDD	KL	DS	MDD	KL	DS	MDD	KL
KoVAE	18.0	57.0	480.4	20.8	6.7	91.4	49.2	3.6	7.1	39.8	4.7	25.1
GT-GAN	35.3	27.3	13.8	31.3	6.1	20.2	49.9	5.2	5.5	49.1	6.0	3.2
Diffusion-NCDE	31.7	16.5	6.5	46.9	5.2	27.7	48.2	3.9	7.9	41.8	3.9	4.0
ProFITi	35.2	12.6	13.0	47.9	3.9	11.2	49.8	3.5	3.4	49.7	3.6	1.8
HeTVAE	49.7	35.2	263.4	41.9	8.0	136.6	49.9	10.1	91.4	49.9	10.5	115.2
Diff-MN	13.8	11.2	2.3	**12.8**	2.8	8.3	46.9	2.7	1.9	35.4	3.2	1.6
HeteroNet	**7.9**	**5.6**	**1.5**	14.2	**1.9**	**7.6**	**34.3**	**1.9**	**1.4**	**26.0**	**2.4**	**0.5**

Results. The aggregate results in Tab. 5 show favorable agreement on MDD and KL across the four datasets. Since generation uses a separately trained latent diffusion model, these scores characterize the complete generation pipeline. Additional results in Appendix D.1 show that HeteroNet's generated samples overlap with real samples in t-SNE projections and capture prominent features of the reference marginal value distributions. Results in Appendix E.5 show that HeteroNet leads most comparisons across the evaluated sequence lengths and training missing rates.

4.2 MODEL ANALYSIS

Ablation Study. We ablate HeteroNet's components to assess their contributions (Fig. 5). Removing Observation Encoding (w/o OE) or Sparse Pattern Tokenization (w/o SPT) degrades performance, highlighting the need to preserve temporal context and explicitly represent complementary states, transitions, and histories. Removing Pattern Interaction (w/o PI) limits information exchange, weakening cross-variate dependency modeling, while omitting Kernel Fusion (w/o KF) discards crucial local value and sampling summaries. For TPP, the prediction head directly uses the sequence representation from Pattern Interaction. Adding Kernel Fusion (w/ KF) increases event-time RMSE. Additional results in Appendix D.2 show that moderate model capacity, a compact transition-token budget, and an intermediate number of temporal kernels yield strong classification performance. Additional results in Appendix D.3 show that HeteroNet outperforms the tested replacement encoders under identical output heads and training objectives.

CKA Similarity Analysis. We examine the relationship between first-to-last-layer representation similarity and task performance using centered kernel alignment (CKA) (Kornblith et al., 2019) (Fig. 6). Across the compared models, stronger performance tends to coincide with lower first-

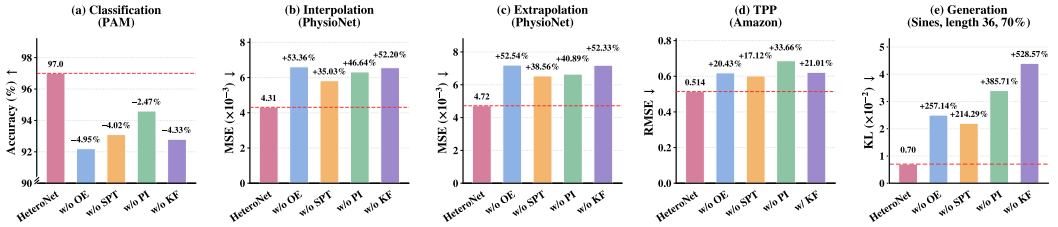


Figure 5: Ablation study of HeteroNet across five tasks. For TPP, w/ KF adds Kernel Fusion before the prediction head. For the other tasks, w/o KF removes this module.

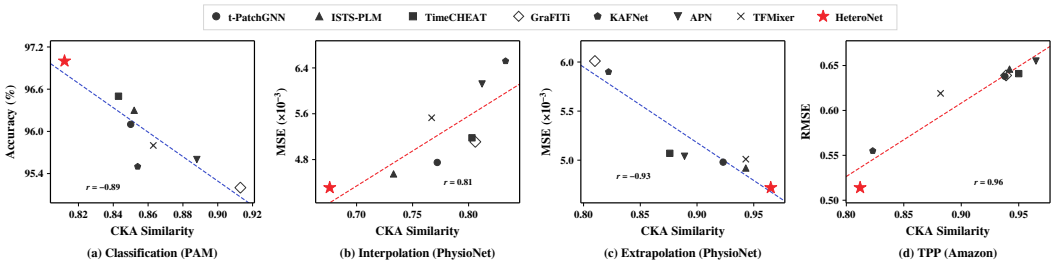


Figure 6: Relationship between the representation similarity and performance across four tasks.

to-last-layer CKA in classification, interpolation, and TPP, but with higher CKA in extrapolation. For classification, greater representational change is consistent with emphasizing class-relevant patterns despite irregular sampling. Interpolation requires integrating asynchronous observations to infer missing values, while Kernel Fusion still supplies local value cues. TPP instead uses the representation from Pattern Interaction, without Kernel Fusion, to predict the next event's time and type. In extrapolation, HeteroNet's higher CKA is consistent with preserving observed states and temporal changes that help anchor predictions beyond the input window. The useful degree of representation change thus depends on what each task needs to predict.

Heterogeneous Pattern Interaction.

Fig. 7 visualizes the patterns and their interactions on a sample from Human Activity (Vidulin et al., 2010). The Sparse Pattern Tokenization module selects recent observations and transitions with the largest elapsed intervals from the asynchronous input. In this example, the observation token Q_1 of variate V_1 assigns relatively high attention to the transition token K_1 of variate V_3. This illustrates how Pattern Interaction relates different token types across asynchronously observed variates. Additional results in Appendix D.4 show that the full-input kernel summaries represent local value patterns and variations in observation concentration across the input window, complementing the selected sparse tokens. Results in Appendix D.5 show that HeteroNet achieves outstanding computational efficiency among the compared methods.

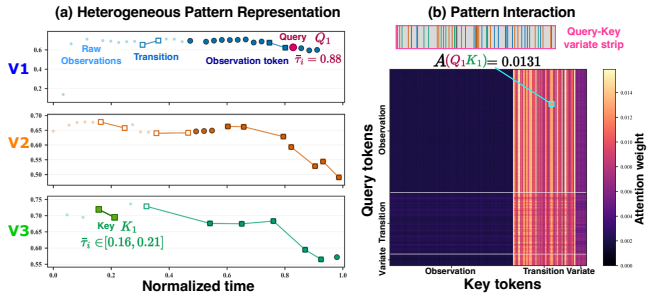


Figure 7: Visualization of pattern representation and interaction. **(a)** Observation and Transition tokens extracted from raw observations. **(b)** 4-layer attention heatmap. Boundary strips indicate token variate identities matching the colors in (a), highlighting the inter-variate pattern interaction.

5 Conclusion and Future Work

This paper presents **HeteroNet**, a TSPM specifically designed for IMTS. HeteroNet models irregular temporal variations and asynchronous cross-variate dependencies through heterogeneous pattern representation and interaction, complemented by temporal kernel summaries of all observations. Extensive experiments on 15 datasets across five IMTS analysis tasks demonstrate its effectiveness against 40 general-purpose and task-specific baselines. We discuss the limitations of HeteroNet in Appendix F. Future work will explore adaptive pattern selection and shared pretraining across datasets and tasks to further improve representation efficiency and transferability.

AI USE STATEMENT

LLMs are used to assist with grammar checking and language refinement.

REFERENCES

Emmanuel Bacry, Martin Bompaire, Stéphane Gaïffas, and Soren Poulsen. Tick: a python library for statistical learning, with a particular emphasis on time-dependent modelling. *arXiv:1707.03003*, 2017.

Marin Bilos, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. In *NeurIPS*, 2021.

Luis Candanedo. Appliances energy prediction. UCI Machine Learning Repository, 2017.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 2018.

Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. *arXiv:2011.04583*, 2020.

Bowen Deng, Chang Xu, Hao Li, Yu-hao Huang, Min Hou, and Jiang Bian. Tardiff: Target-oriented diffusion guidance for synthetic electronic health record time series generation. In *SIGKDD*, 2025.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv:2111.08095*, 2021.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *SIGKDD*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten M. Borgwardt. Set functions for time series. In *ICML*, 2020.

Yiming Huang, Ziyu Zhou, Zihao Wang, et al. Timesnet-pm2.5: Interpretable timesnet for disentangling intraperiod and interperiod variations in pm2.5 prediction. *Atmosphere*, 2023.

Yu-Hao Huang, Chang Xu, Yueying Wu, Wu-Jun Li, and Jiang Bian. Timedp: Learning to generate multi-domain time series with domain prompts. In *AAAI*, 2025.

Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. In *NeurIPS*, 2022.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *ICLR*, 2024.

Alistair E. W. 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*, 2016.

Patrick Kidger, James Morrill, James Foster, and Terry J. Lyons. Neural controlled differential equations for irregular time series. In *NeurIPS*, 2020.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. Hyperimts: Hypergraph neural network for irregular multivariate time series forecasting. In *ICML*, 2025.

Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. In *NeurIPS*, 2023.

Jiexi Liu, Meng Cao, and Songcan Chen. Musicnet: A gradual coarse-to-fine framework for irregularly sampled multivariate time series analysis. arXiv:2412.01063, 2024.

Jiexi Liu, Meng Cao, and Songcan Chen. Timecheat: A channel harmony strategy for irregularly sampled multivariate time series analysis. In *AAAI*, 2025a.

Jiexi Liu, Meng Cao, and Songcan Chen. Beyond observations: Reconstruction error-guided irregularly sampled time series representation learning. In *AAAI*, 2026a.

Qinghua Liu, Sam Heshmati, Zheda Mai, Zubin Abraham, John Paparrizos, and Liu Ren. Mllm4ts: Leveraging vision and multimodal language models for general time-series analysis. *arXiv:2510.07513*, 2025b.

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*, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

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

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-patch: Hierarchical patch gnn for irregular multivariate time series. In *ICML*, 2025.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NeurIPS*, 2017.

M. J. Menne, C. N. Jr. Williams, and R. S. Vose. Long-term daily and monthly climate records from stations across the contiguous united states (u.s. historical climatology network). *United States Historical Climatology Network Dataset*, 2016.

Ilan Naiman, N. Benjamin Erichson, Pu Ren, Michael W. Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via koopman VAEs. In *ICLR*, 2024.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*, 2019.

Takahiro Omi, naonori ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In *NeurIPS*, 2019.

Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. In *ICML*, 2026.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, 2012.

Matthew Reyna, Chris Josef, Russell Jeter, Supreeth Shashikumar, Benjamin Moody, 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. *PhysioNet*, 2019.

Weilin Ruan, Xilin Dang, Ziyu Zhou, Sisuo Lyu, and Yuxuan Liang. A retrieval augmented spatio-temporal framework for traffic prediction. In *AAAI*, 2026.

Yulia Rubanova, Tian Qi Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, 2019.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *ICML*, 2022.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. *arXiv:1909.12127*, 2019.

Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T. Kwok. Time series anomaly detection with multiresolution ensemble decoding. In *AAAI*, 2021.

Lifeng Shen, Weiyu Chen, and James Kwok. Multi-resolution diffusion models for time series forecasting. In *ICLR*, 2024.

Lifeng Shen, Kai Syun Hou, Weiyu Chen, and James Kwok. Latent-to-data cascaded diffusion models for unconditional time series generation. In *ICLR*, 2026.

Satya Narayan Shukla and Benjamin Marlin. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *ICLR*, 2022.

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

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *ICLR*, 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 *Computing in Cardiology*, 2012.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: a physics engine for model-based control. In *IROS*, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Vedrana Vidulin, Mitja Lustrek, Bostjan Kaluza, Rok Piltaver, and Jana Krivec. Localization Data for Person Activity. UCI Machine Learning Repository, 2010. DOI: https://doi.org/10.24432/C57G8X.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *ICLR*, 2025.

Yanyun Wang, Dehui Du, Haibo Hu, Zi Liang, and Yuanhao Liu. Tsfool: Crafting highly-imperceptible adversarial time series through multi-objective attack. arXiv:2209.06388, 2024.

Chris Whong. FOILing NYC's taxi trip data, 2014. URL `https://chriswhong.com/open-data/foil_nyc_taxi/`.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021a.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 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 *AAAI*, 2021b.

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 *SIGKDD*, 2020.

Tao Xie, Zexi Tan, Haoyi Xiao, Mengke Li, Yiqun Zhang, Yang Lu, Cuie Yang, and Yiu-ming Cheung. Anchormoe: Interpretable time series classification via anchor-routed moe. In *SIGKDD*, 2026.

Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. In *NeurIPS*, 2022.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Chen Pan, James Zhang, Qingsong Wen, JUN ZHOU, et al. Easytpp: Towards open benchmarking temporal point processes. In *ICLR*, 2024.

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 *AAAI*, 2024.

Vijaya Krishna Yalavarthi, Randolf Scholz, Stefan Born, and Lars Schmidt-Thieme. Probabilistic forecasting of irregularly sampled time series with missing values via conditional normalizing flows. In *AAAI*, 2025.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. *arXiv:2201.00044*, 2021.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *NeurIPS*, 2019.

Da Zhang, Bingyu Li, Zhiyuan Zhao, Feiping Nie, Junyu Gao, and Xuelong Li. Fusad: Time-frequency fusion with adaptive denoising for general time series analysis. In *ICDE*, 2026a.

Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *SIGKDD*, 2023.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *ICML*, 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 *ICML*, 2024a.

Weijia Zhang, Le Zhang, Jindong Han, Hao Liu, Yanjie Fu, Jingbo Zhou, Yu Mei, and Hui Xiong. Irregular traffic time series forecasting based on asynchronous spatio-temporal graph convolutional networks. In *SIGKDD*, 2024b.

Weijia Zhang, Chenlong Yin, Hao Liu, and Hui Xiong. Unleashing the power of pre-trained language models for irregularly sampled time series. In *SIGKDD*, 2025.

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

Xu Zhang, Junwei Deng, Chang Xu, Hao Li, and Jiang Bian. Diff-mn: Diffusion parameterized moe-ncde for continuous time series generation with irregular observations. In *ICML*, 2026b.

Shuhan Zhong, Sizhe Song, Weipeng Zhuo, Guanyao Li, Yang Liu, and S.-H. Gary Chan. A multi-scale decomposition mlp-mixer for time series analysis. In *Proc. VLDB Endow.*, 2024.

Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *ICML*, 2013.

Tian Zhou, Peisong Niu, xue wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

Wang-Tao Zhou, Zhao Kang, Ke Yan, and Ling Tian. Itpp: Learning disentangled event dynamics in marked temporal point processes. In *AAAI*, 2026a.

Ziyu Zhou, Gengyu Lyu, Yiming Huang, Zihao Wang, Ziyu Jia, and Zhen Yang. Sdformer: Transformer with spectral filter and dynamic attention for multivariate time series long-term forecasting. In *IJCAI*, 2024.

Ziyu Zhou, Jiaxi Hu, Qingsong Wen, James T. Kwok, and Yuxuan Liang. Multi-order wavelet derivative transform for deep time series forecasting. In *CIKM*, 2026b.

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*, 2026c.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *ICML*, 2020.

Appendix (Supplementary Material)

TABLE OF CONTENTS

A Related Works **15**

A.1 Time Series Pattern Machines 15

A.2 Irregular Time Series Analysis 15

B Output-head **15**

C Experimental Details **16**

C.1 Dataset 16

C.2 Baselines 17

C.3 Evaluation Metrics 17

C.4 Implementation Details 18

D Model Analysis **19**

D.1 Qualitative Evaluation for Generation 19

D.2 Hyperparameter Sensitivity 19

D.3 Controlled Backbone and Head Ablations 20

D.4 Complementary Kernel Summaries 21

D.5 Efficiency Analysis 21

E Complete Experimental Results **22**

E.1 Classification 22

E.2 Interpolation 22

E.3 Extrapolation 23

E.4 Temporal Point Process 23

E.5 Generation 24

F Limitations **24**

A Related Works

A.1 Time Series Pattern Machines

Time series pattern machines aim to support diverse analysis tasks within a common architecture. These tasks include forecasting (Wu et al., 2021a; 2023; Zhou et al., 2024), classification (Xie et al., 2026), imputation (Tashiro et al., 2021), and anomaly detection (Shen et al., 2021). For general time series analysis, TimesNet (Wu et al., 2023) models intraperiod and interperiod variations through two-dimensional representations, ModernTCN (Luo & Wang, 2024) develops a convolutional backbone with a large effective receptive field, and MSD-Mixer (Zhong et al., 2024) learns multi-scale decomposition using MLPs. TimeMixer++ (Wang et al., 2025) combines patterns across scales and resolutions, while FusAD (Zhang et al., 2026a) integrates time-frequency fusion with adaptive denoising. Pretrained language and multimodal models provide another route to general analysis by transferring representations across modalities (Zhou et al., 2023; Liu et al., 2025b). However, these architectures primarily organize patterns on regular time grids, leaving uneven sampling and cross-variate asynchrony insufficiently addressed, which motivates HeteroNet's explicit representation and interaction of heterogeneous patterns in IMTS.

A.2 Irregular Time Series Analysis

Existing IMTS methods address missing observations and uneven intervals through recurrent models with learned decay (Che et al., 2018) and continuous-time dynamics (Rubanova et al., 2019; Kidger et al., 2020). Attention-based approaches incorporate temporal information and representations at multiple scales (Shukla & Marlin, 2021; Zhang et al., 2023), while graph-based methods model temporal and cross-variate dependencies (Zhang et al., 2022; Yalavarthi et al., 2024; Zhang et al., 2024a; Li et al., 2025). Toward general IMTS analysis, ISTS-PLM (Zhang et al., 2025) adapts pretrained language models for temporal and variate representation learning, while TimeCHEAT (Liu et al., 2025a) combines local channel dependence with global channel independence, supporting classification, interpolation, and extrapolation. Complementary directions include temporal point process modeling for event timing and types, benchmarked by EasyTPP (Xue et al., 2024), and generation from irregular observations, for which Diff-MN (Zhang et al., 2026b) combines diffusion with mixture-of-experts neural controlled differential equations. Despite these advances, explicitly relating observed states, temporal transitions, and variate histories within a common backbone for both predictive and generative IMTS tasks remains underexplored, which HeteroNet addresses through heterogeneous pattern representation and interaction.

B Output-head

Fig. 8 illustrates how the task-specific heads convert the learned representations into task outputs. Following Sec. 3, $\mathbf{Z}$ denotes the contextualized pattern tokens, while $\mathbf{H}$ and $\mathbf{h}$ are the final variate-level and sequence-level representations. We write $\mathbf{h}_c$ for the representation of variate c in the c-th row of $\mathbf{H}$. Each task is trained separately.

Classification. We follow the implementations (Zhang et al., 2023; 2025). Mean and max pooling over the variate representations $\mathbf{H}$ produce a fixed-length summary. We concatenate this summary with $\mathbf{h}$, project it through an MLP, and add the result to $\mathbf{h}$. A classification MLP maps the fused features to class logits. Softmax yields class probabilities, and argmax gives the predicted label $\widehat{y}$.

Interpolation and extrapolation. We follow the implementations (Zhang et al., 2024a; 2025; Liu et al., 2025a). These tasks share a value prediction head. For query $(t_j, c_j) \in \mathcal{Q}$, the head combines the target-time embedding with $\mathbf{h}_{c_j}$ to attend to $\mathbf{Z}$, accounting for temporal distance and variate identity. Query times use the same normalization as the observed timestamps. An MLP combines the retrieved context with $\mathbf{h}$, $\mathbf{h}_{c_j}$, and the target-time embedding to produce a scalar prediction. A learnable weighted sum combines this prediction with a local kernel estimate from observations of variate c_j, yielding $\widehat{x}_j$. Collecting the predictions gives $\widehat{\mathcal{X}} = \{\widehat{x}_j\}_{j=1}^{|\mathcal{Q}|}$. Interpolation queries lie within the observed window; extrapolation queries satisfy $t_j > \max_i \tau_i$.

Temporal point process modeling. We follow the implementation (Xue et al., 2024). Let $\mathbf{h}_{1:i}^{\mathrm{tok}}$ denote the sequence-level representation produced by Pattern Interaction from the causal history window

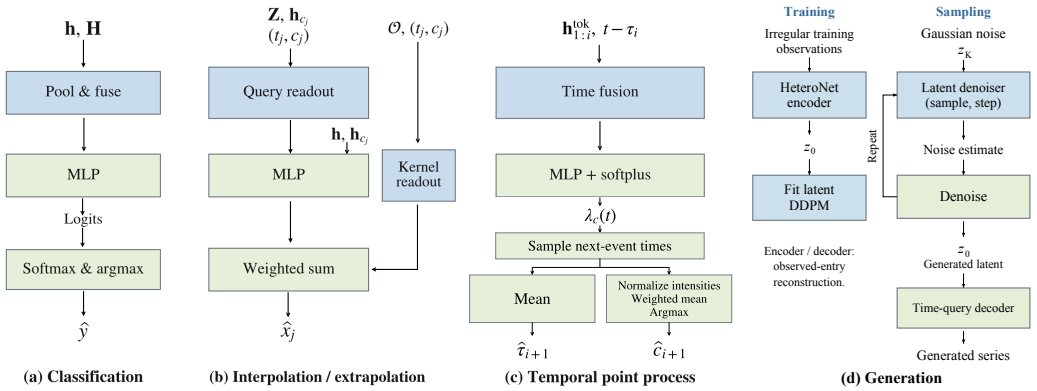


Figure 8: Task-specific output heads of HeteroNet. (a) Classification pools $\mathbf{H}$ and fuses the pooled features with $\mathbf{h}$ to predict $\widehat{y}$. (b) Interpolation and extrapolation predict $\widehat{x}_j$ for each query (t_j, c_j) within or after the observed window, respectively. (c) TPP predicts the next timestamp $\widehat{\tau}_{i+1}$ and event type $\widehat{c}_{i+1}$ from history-conditioned intensities, with type probabilities estimated over sampled next-event times. (d) Generation fits a diffusion model to HeteroNet's training latents. Starting from Gaussian noise, the trained model samples a new latent and a time-query decoder produces the generated sequence.

ending at event i. The intensity head combines this representation with an embedding of the elapsed time $t - \tau_i$. An MLP followed by softplus produces a positive intensity $\lambda_c(t)$ for each event type, with dependence on $\mathcal{O}_{1:i}$ left implicit. Together, these intensities specify the conditional distribution $p_\theta(\tau_{i+1}, c_{i+1} \mid \mathcal{O}_{1:i})$. The predicted timestamp $\widehat{\tau}_{i+1}$ is obtained by averaging sampled waiting times and adding τ_i. The predicted type is the argmax of the weighted average of normalized type intensities evaluated at the sampled next-event times. Training minimizes the negative log-likelihood of the event sequence.

Generation. We use a latent diffusion head for time-series generation. During training, HeteroNet encodes each irregular series $\mathcal{O}^{(n)}$ into a latent vector $\mathbf{z}_0^{(n)} = E_\phi(\mathcal{O}^{(n)})$. The encoder pools the variate representations $\mathbf{H}$, fuses them with the sequence representation $\mathbf{h}$, and projects the result to the latent space. A time-query decoder $D_\omega(\mathbf{z}, \mathcal{Q})$ maps a latent vector and output-time embeddings to multivariate values. The encoder and decoder are first trained using reconstruction loss at observed training entries only. We then freeze them and train a diffusion model on the encoded training latents. The latent denoiser receives the noisy latent and diffusion-step embedding. At sampling time, reverse diffusion starts from Gaussian noise, produces a new latent vector, and decodes it onto $\mathcal{Q}$. No evaluation-sequence observations or observation masks are supplied to this sampling procedure.

C Experimental Details

C.1 Dataset

For classification, we use PhysioNet 2012 (P12) (Silva et al., 2012), PhysioNet 2019 (P19) (Reyna et al., 2019), and PAM (Reiss & Stricker, 2012) for mortality prediction, sepsis detection, and human activity recognition, respectively. After preprocessing, they contain 11,988, 38,803, and 5,333 sequences with 36, 34, and 17 dynamic variates, respectively. Their missing rates are 88.4%, 94.9%, and 60%. The training, validation, and test splits follow an 80%/10%/10% ratio.

For interpolation and extrapolation, we use PhysioNet (Silva et al., 2012) and MIMIC (Johnson et al., 2016) from healthcare, Human Activity (Vidulin et al., 2010) from motion sensing, and USHCN (Menne et al., 2016) from climate monitoring. They contain 41, 96, 12, and 5 variates, with missing rates of 85.7%, 96.7%, 75%, and 95%, respectively, and use 60%/20%/20% training/validation/test splits. The two tasks query values within and after the observed window, respectively.

For temporal point process modeling, we use Amazon (Ni et al., 2019), Retweet (Zhou et al., 2013), Taxi (Whong, 2014), and Taobao (Xue et al., 2022) from the EasyTPP benchmark (Xue et al., 2024). These datasets describe product reviews, retweet cascades, taxi trips, and online shopping interactions,

with 16, 3, 10, and 17 event types, respectively. Each sequence records event timestamps and types. We adopt the provided data splits and evaluate the prediction of the next event's time and type.

For generation, we use Sines (Yoon et al., 2019), Stocks (Yoon et al., 2019), Energy (Candanedo, 2017), and MuJoCo (Todorov et al., 2012), containing 5, 6, 28, and 14 variates, respectively. We follow the irregular-to-regular generation benchmark of Diff-MN (Zhang et al., 2026b), using sequence lengths $\{12, 24, 36\}$ and observation-dropping rates $\{30\%, 50\%, 70\%\}$. The dropping rate specifies the incompleteness of the observations used to learn the generator. Generated sequences are sampled on the full regular grid and evaluated against the benchmark reference collection. They are not paired completions of partially observed evaluation sequences.

C.2 Baselines

We compare HeteroNet with 40 baselines across five tasks. The shared comparison includes t-PatchGNN (Zhang et al., 2024a), ISTS-PLM (Zhang et al., 2025), TimeCHEAT (Liu et al., 2025a), GraFITi (Yalavarthi et al., 2024), KAFNet (Zhou et al., 2026c), APN (Liu et al., 2026b), and TFMixer (Qiu et al., 2026) across all five tasks, together with Trans-mean (the vanilla Transformer (Vaswani et al., 2017) with mean imputation for missing values) , GRU-D (Che et al., 2018), SeFT (Horn et al., 2020), mTAND (Shukla & Marlin, 2021), Raindrop (Zhang et al., 2022), Warpformer (Zhang et al., 2023), and Latent-ODE (Rubanova et al., 2019) across classification, interpolation, and extrapolation, following (Zhang et al., 2025). Additional comparisons include Transformer (Vaswani et al., 2017), MTGNN (Wu et al., 2020), DGM^2-O (Wu et al., 2021b) , IP-Net (Shukla & Marlin, 2019), FPT (Zhou et al., 2023), Time-LLM (Jin et al., 2024), and ViTST (Li et al., 2023) for classification, and Neural Flow (Bilos et al., 2021) and CRU (Schirmer et al., 2022) for interpolation and extrapolation. For temporal point process modeling, we additionally compare with task-specific baselines: MHP (Bacry et al., 2017), RMTPP (Du et al., 2016), NHP (Mei & Eisner, 2017), SAHP (Zhang et al., 2020), THP (Zuo et al., 2020), AttNHP (Yang et al., 2021), ODETPP (Chen et al., 2020), FullyNN (Omi et al., 2019), and IFTPP (Shchur et al., 2019), following (Xue et al., 2024). For generation, the additional task-specific baselines are KoVAE (Naiman et al., 2024), GT-GAN (Jeon et al., 2022), TimeGAN-NCDE (Yoon et al., 2019), TimeVAE-NCDE (Desai et al., 2021), Diffusion-NCDE (Huang et al., 2025), ProFITi (Yalavarthi et al., 2025), HeTVAE (Shukla & Marlin, 2022) , and Diff-MN (Zhang et al., 2026b). For methods and settings covered by the published comparisons, baseline results are taken from ISTS-PLM (Zhang et al., 2025) for classification, interpolation, and extrapolation, EasyTPP (Xue et al., 2024) for the dedicated TPP baselines, and Diff-MN (Zhang et al., 2026b) for the dedicated generation baselines. Additional general-purpose baseline results are obtained from our own reproductions. TPP entries not reported in EasyTPP (Xue et al., 2024) are marked as "–"; its event-type error means and standard deviations are expressed in percentage points here.

C.3 Evaluation Metrics

Higher values are better for classification metrics; lower values are better for the remaining metrics.

Classification. For N test samples and K_{cls} classes, let y_i and $\widehat{y}_i$ denote true and predicted labels. PAM uses accuracy and macro-averaged precision, recall, and F1:

$$\begin{aligned} \text{Accuracy} &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[\widehat{y}_i = y_i], & \text{Precision} &= \frac{1}{K_{\text{cls}}}\sum_{k=1}^{K_{\text{cls}}}\frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}, \\ \text{Recall} &= \frac{1}{K_{\text{cls}}}\sum_{k=1}^{K_{\text{cls}}}\frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}, & \text{F1} &= \frac{1}{K_{\text{cls}}}\sum_{k=1}^{K_{\text{cls}}}\frac{2\text{TP}_k}{2\text{TP}_k + \text{FP}_k + \text{FN}_k}. \end{aligned} \tag{9}$$

Here, TP_k, FP_k, and FN_k are the classwise true-positive, false-positive, and false-negative counts. Zero-denominator terms are set to zero. P12 and P19 use AUROC and AUPRC, with AUPRC computed as average precision:

$$\text{AUROC} = \int_0^1 \text{TPR}(u)\,\mathrm{d}u, \tag{10}$$

$$\text{AUPRC} = \sum_{q=1}^{J}(R_q - R_{q-1})P_q, \qquad R_0 = 0. \tag{11}$$

Here, $\mathrm{TPR}(u)$ is the true-positive rate at false-positive rate u, and P_q, R_q are precision and recall at decreasing score thresholds. Classification scores are reported as percentages.

Interpolation and extrapolation. Let N_c be the number of valid test targets for variate c and $\mathcal{C}_+ = \{c : N_c > 0\}$. Following the IMTS evaluation protocol, MSE and MAE average valid errors within each variate and then across variates:

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{|\mathcal{C}_+|} \sum_{c \in \mathcal{C}_+} \frac{1}{N_c} \sum_{j=1}^{N_c} (\widehat{x}_{jc} - x_{jc})^2, \\ \mathrm{MAE} &= \frac{1}{|\mathcal{C}_+|} \sum_{c \in \mathcal{C}_+} \frac{1}{N_c} \sum_{j=1}^{N_c} |\widehat{x}_{jc} - x_{jc}|, \end{aligned} \tag{12}$$

where x_{jc} and $\widehat{x}_{jc}$ are true and predicted target values.

Temporal point process modeling. Following EasyTPP (Xue et al., 2024), we evaluate N_e valid next-event predictions using time RMSE and type error rate:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_e} \sum_{i=1}^{N_e} (\widehat{\tau}_i - \tau_i)^2}, \quad \mathrm{Err}(\%) = \frac{100}{N_e} \sum_{i=1}^{N_e} \mathbb{I}[\widehat{c}_i \neq c_i], \tag{13}$$

where (τ_i, c_i) and $(\widehat{\tau}_i, \widehat{c}_i)$ denote true and predicted event times and types. Let $\mathbb{I}[\cdot]$ denote the indicator function. Type predictions are obtained by taking the argmax of the weighted average of normalized type intensities over sampled next-event times.

Generation. Following Diff-MN (Zhang et al., 2026b), we report discriminative score (DS), marginal distribution difference (MDD), and Kullback–Leibler divergence (KL):

$$\mathrm{DS} = \left| a_{\mathrm{disc}} - \frac{1}{2} \right|, \tag{14}$$

$$\mathrm{MDD} = \frac{1}{TCB} \sum_{t=1}^{T} \sum_{c=1}^{C} \sum_{b=1}^{B} |f_{tcb} - \widetilde{f}_{tcb}|, \tag{15}$$

$$\mathrm{KL} = \sum_{b=1}^{B_{\mathrm{KL}}} p_b \log \frac{p_b}{q_b}. \tag{16}$$

Here, a_{disc} is a real-versus-generated discriminator's accuracy on a balanced held-out set. MDD compares real and generated histogram densities f_{tcb} and $\widetilde{f}_{tcb}$ using $B = 20$ shared bins at each of T time positions and C variates. For KL, p_b, q_b are normalized histograms of pooled real and generated values using $B_{\mathrm{KL}} = 50$ shared bins, with 10^{-9} added to generated densities before normalization.

C.4 Implementation Details

We implement HeteroNet in PyTorch on a single NVIDIA H100 GPU with 80 GB memory. Task protocols follow ISTS-PLM (Zhang et al., 2025) for classification, interpolation, and extrapolation, EasyTPP (Xue et al., 2024) for TPP, and Diff-MN (Zhang et al., 2026b) for generation. Batch sizes are 32 for classification/interpolation/extrapolation and 128 for TPP/generation. We use Adam (Kingma & Ba, 2014) for classification and TPP, and AdamW (Loshchilov & Hutter, 2019) otherwise. Learning rates are 4×10^{-4} for P12 classification, 10^{-3} for generation, and 5×10^{-4} otherwise. For classification, interpolation, extrapolation, and TPP, HeteroNet's reported means and standard deviations are computed over three runs.

The IMTS loaders pack observations by variate and zero-pad them. L_{in} denotes the padded input length before tokenization. Profile density is the observed count divided by this length. Masks exclude unavailable entries from value statistics and valid timestamp extrema. Unobserved variates retain zero-count profile tokens, while unavailable event/transition tokens are masked. Empty masked-max pools return zero; degenerate time ranges use a unit denominator, and mean, slope, and kernel denominators use 10^{-8} stabilization. Backbone time features describe relative within-window timing; regression uses supplied query timestamps.

Tasks are trained separately using class-weighted cross-entropy, masked MSE (Huber loss on USHCN), or TPP negative log-likelihood. For TPP, the last at most W events of $\mathcal{O}_{1:i}$ are selected before time normalization, token selection, and profile computation. Padded events are masked, and no full-sequence features are cached. The TPP benchmarks contain event times and categorical types, without numerical measurement amplitudes. Each event is encoded as a unit-valued observation in its type channel. For observed types, the kernel-weighted value averages μ_{ck} in Eq. 6 therefore reduce to one before numerical stabilization, although ρ_{ck} can still summarize event timing. The TPP output layer directly maps the sequence-level representation from Pattern Interaction to conditional intensities, without Kernel Fusion, following the history-to-intensity formulation of neural TPP models (Mei & Eisner, 2017; Zuo et al., 2020). Its intensity head and likelihood retain dataset time units. Target events enter the likelihood but not history encoding.

For generation, observed-entry reconstruction MSE trains the encoder and time-query decoder. Both are then frozen, and a latent DDPM (Ho et al., 2020) learns standardized training latents using noise-prediction MSE. Sampling starts from Gaussian noise; generated latents are inverse-standardized and decoded on a regular normalized grid, without evaluation observations or marginal calibration.

D Model Analysis

D.1 Qualitative Evaluation for Generation

Following (Zhang et al., 2026b), Fig. 9 compares real and generated sequences through two-dimensional t-SNE embeddings and marginal probability density functions. From left to right, we evaluate on Sines, Stocks, Energy, and MuJoCo, with a missing rate of 70%. HeteroNet's generated samples overlap with the real samples in the displayed projections, while its density curves show agreement with prominent features of the reference distributions. These visualizations complement the quantitative evaluation by illustrating projected sequence distributions and marginal value distributions across datasets.

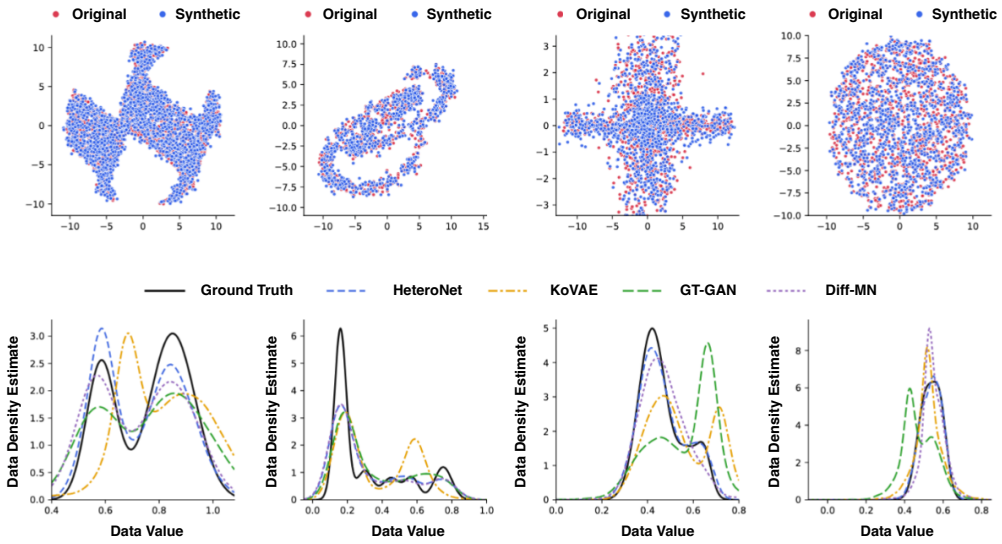


Figure 9: Qualitative comparison of real and generated time series. **Top:** Two-dimensional t-SNE embeddings of real sequences and samples generated by HeteroNet. **Bottom:** Marginal value densities of real data and samples from HeteroNet and representative generative baselines (Naiman et al., 2024; Jeon et al., 2022; Zhang et al., 2026b).

D.2 Hyperparameter Sensitivity

Fig. 10 shows that HeteroNet performs best with moderate model capacity. Increasing d up to 128 improves Accuracy, Precision, and F1, whereas $d = 256$ causes a clear Accuracy drop. These results favor a moderate representation dimension among the settings evaluated on PAM. Two interaction

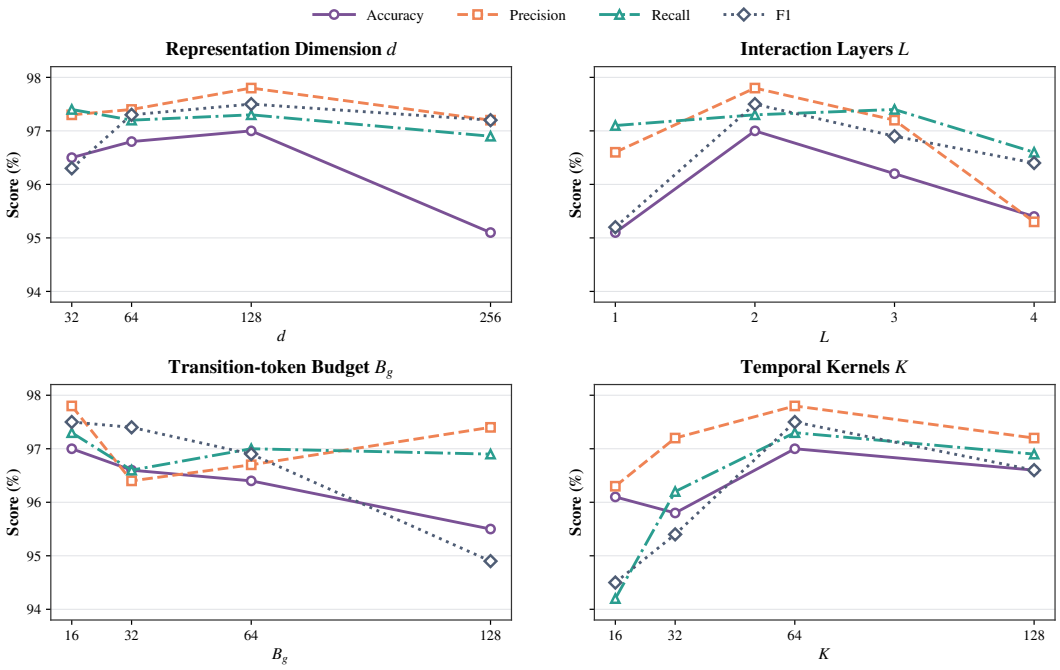


Figure 10: Hyperparameter sensitivity of HeteroNet on PAM. We vary the representation dimension d, the number of Pattern Interaction layers L, the transition-token budget B_g, and the number of temporal kernels K, while keeping the remaining settings fixed. All metrics are reported in percentage points, and higher values are better.

layers provide the strongest overall balance among the tested settings, whereas four layers reduce every metric. The preferred interaction depth may differ across tasks and datasets.

The tokenization and kernel results also favor intermediate configurations on PAM. A transition budget of $B_g = 16$ provides strong overall performance, while increasing it to 128 reduces F1. Increasing K from 16 to 64 improves Recall and F1, followed by a decline at $K = 128$. These results guide the choice of token budget and kernel resolution for this dataset.

D.3 Controlled Backbone and Head Ablations

We conduct controlled comparisons to distinguish the contributions of HeteroNet's backbone, output heads, and training objectives (Tab. 6). Encoder replacements use identical output-head architectures and training objectives, with data splits, preprocessing, training budgets, and evaluation protocols held fixed. Additional controls isolate the token and kernel branches, remove individual token types, and disable the prediction head's direct kernel estimate. This estimate is specific to interpolation and extrapolation and is distinct from the backbone's Kernel Fusion branch. Individual token removals are compared with a full-backbone control using the same variate-pooling readout. Generation is evaluated on Energy with sequence length 36 and a training missing rate of 50%. We also compare MSE and Huber training losses on USHCN. All entries report point estimates.

With identical heads and training objectives, HeteroNet reduces interpolation MSE, extrapolation MSE, and generation KL by **6.1%**, **24.2%**, and **9.1%**, respectively, relative to the strongest alternative encoder in each setting, supporting the contribution of its heterogeneous pattern representations. Disabling the prediction head's direct kernel estimate increases interpolation MSE and extrapolation MSE, showing that this head component also contributes to accuracy. Without this estimate, combining the token and kernel branches does not consistently outperform either branch alone, while individual token removals under the shared readout produce smaller, task-dependent changes. On USHCN, HeteroNet outperforms the Observation Transformer under both training objectives. Replacing Huber with MSE changes its interpolation MSE slightly and leaves extrapolation MSE

Table 6: Controlled ablations of HeteroNet. (A) Component removal. (B) Encoder replacement with identical heads and objectives. (C) Representation and prediction-head controls. (D) Training-loss comparison on USHCN. Lower is better for all metrics. — indicates that the direct kernel estimate is not used by the generation head.

Variant	Interpolation PhysioNet MSE ($\times 10^{-3}$)	Extrapolation PhysioNet MSE ($\times 10^{-3}$)	Generation Energy (36, 50%) KL ($\times 10^{-2}$)
A. Existing results			
HeteroNet	4.31	4.72	1.50
w/o Observation Encoding	6.61	7.20	1.92
w/o Sparse Pattern Tokenization	5.82	6.54	2.03
w/o Pattern Interaction	6.32	6.65	2.12
w/o Kernel Fusion	6.56	7.19	2.46
B. Encoder replacement with identical heads and objectives			
Mean-pooling MLP	4.59	6.23	1.65
Observation Transformer	4.73	7.77	2.02
KAFNet encoder	5.55	7.12	1.71
TimeCHEAT encoder	5.73	6.63	1.69
C. Representation and head			
Token branch only, head kernel off	4.78	5.03	1.91
Kernel branch only, head kernel off	4.82	5.12	1.99
Full backbone, head kernel off	5.21	5.08	—
Full backbone, shared variate readout	4.43	4.89	1.68
w/o Observation tokens	4.45	4.87	1.63
w/o Transition tokens	4.48	4.88	1.69
w/o Variate-profile tokens	4.44	4.89	1.67

D. USHCN loss control with identical prediction heads

Backbone	Training loss	Interpolation MSE ($\times 10^{-1}$)	Extrapolation MSE ($\times 10^{-1}$)
HeteroNet (reported)	Huber	5.06	5.04
Observation Transformer	MSE	5.22	5.23
HeteroNet	MSE	5.07	5.04
Observation Transformer	Huber	5.26	5.28

unchanged. These comparisons indicate that its advantage over the alternative encoders cannot be explained solely by the output-head architecture or the use of Huber loss.

D.4 Complementary Kernel Summaries

Fig. 11 visualizes the temporal kernel summaries computed from all valid observations in three examples. The local values μ summarize observed values around each kernel center, while the observation concentration ρ summarizes the distribution of observation times through average kernel weights. Together, they illustrate the local value and sampling information provided by the full-input branch alongside the selected pattern tokens.

D.5 Efficiency Analysis

As shown in Fig. 12, HeteroNet records the shortest inference time and the second-shortest training time among the compared methods in the profiled setting. Its parameter count and FLOPs are smaller than those of most competitors in this comparison, although KAFNet has lower values on both measures. Together with the corresponding predictive results, these measurements indicate a favorable balance between effectiveness and efficiency for this setting. Model size varies with the chosen architecture, while FLOPs also depend on input size; the measurements in this figure do not represent fixed values across all tasks and datasets.

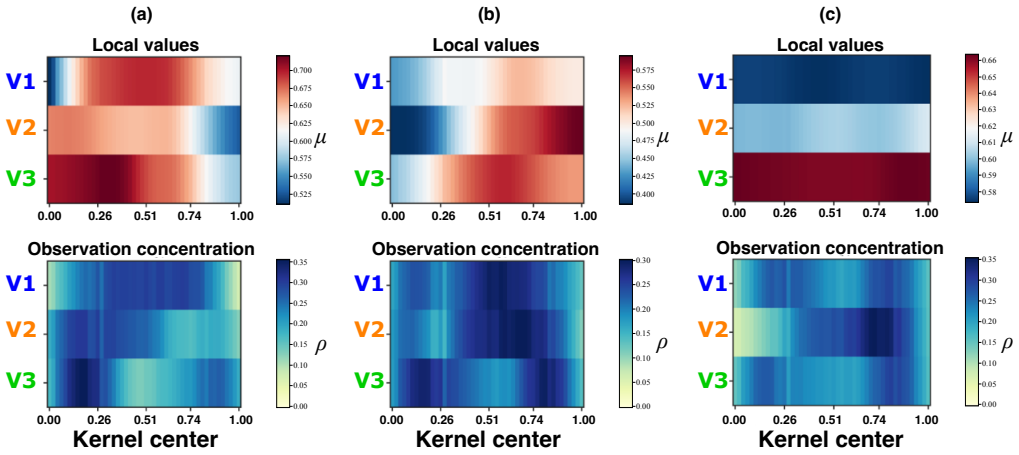
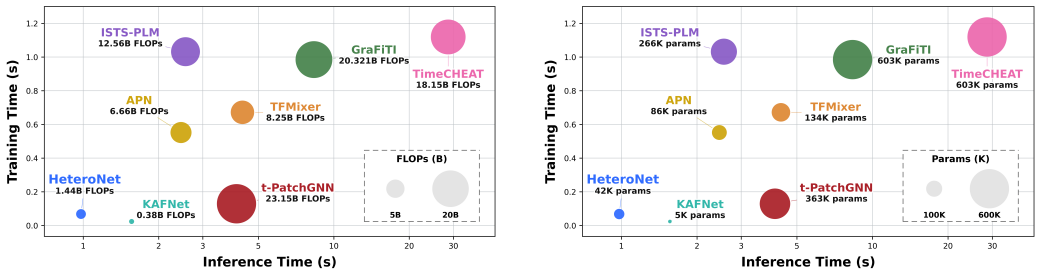


Figure 11: Temporal kernel summaries for three input examples. The top row shows kernel-weighted local values μ. The bottom row shows observation concentration ρ, computed as the average kernel weight over observations of each variate. Columns correspond to kernel centers in normalized time.



(a) Computational cost measured by FLOPs.

(b) Model size measured by trainable parameters.

Figure 12: Efficiency comparison on the MIMIC extrapolation task. All measurements use a batch size of 32. The horizontal and vertical axes show inference time per sample and training time per step, respectively. Bubble size represents FLOPs in (a) and the number of trainable parameters in (b).

E Complete Experimental Results

Tab. 7, Tab. 8, Tab. 9, and Tab. 10 extend the comparisons in Tab. 1, Tab. 2, Tab. 3, and Tab. 4, respectively, with additional baselines. Tab. 11, Tab. 12, and Tab. 13 supplement the aggregate generation results in Tab. 5 with results for sequence lengths 12, 24, and 36, respectively, at each missing rate, including additional baselines.

E.1 Classification

Tab. 7 extends the classification comparison to additional baselines. HeteroNet has the highest AUROC and AUPRC on P19 and the highest accuracy, precision, recall, and F1 score on PAM. On P12, its AUROC is second to ISTS-PLM, while its AUPRC is below ISTS-PLM (Zhang et al., 2025), t-PatchGNN (Zhang et al., 2024a), and TFMixer (Qiu et al., 2026). The P12 results therefore vary by metric, while the P19 and PAM gains hold across all reported metrics for those datasets.

E.2 Interpolation

Tab. 8 shows that HeteroNet obtains the lowest MSE and MAE on all four datasets in the expanded comparison. On MIMIC, its MSE is 1.12, compared with 1.42 for the closest baseline, TimeCHEAT (Liu et al., 2025a). Its leading MSE and MAE results on both Human Activity and

Table 7: Complete classification results (mean ± std). Best and second-best results over all methods are shown in **red** and blue, respectively. Higher values are better. We use results reported in (Zhang et al., 2025) where available and reproduce the remaining baselines.

Method	P12		P19		PAM			
	AUROC	AUPRC	AUROC	AUPRC	Accuracy	Precision	Recall	F1 score
Transformer	83.3 ± 0.7	47.9 ± 3.6	80.7 ± 3.8	42.7 ± 7.7	83.5 ± 1.5	84.8 ± 1.5	86.0 ± 1.2	85.0 ± 1.3
Trans-mean	82.6 ± 2.0	46.3 ± 4.0	83.7 ± 1.8	45.8 ± 3.2	83.7 ± 2.3	84.9 ± 2.6	86.4 ± 2.1	85.1 ± 2.4
MTGNN	74.4 ± 6.7	35.5 ± 6.0	81.9 ± 6.2	39.9 ± 8.9	83.4 ± 1.9	85.2 ± 1.7	86.1 ± 1.9	85.9 ± 2.4
DGM^2-O	84.4 ± 1.6	47.3 ± 3.6	86.7 ± 3.4	44.7 ± 11.7	82.4 ± 2.3	85.2 ± 1.2	83.9 ± 2.3	84.3 ± 1.8
IP-Net	82.6 ± 1.4	47.6 ± 3.1	84.6 ± 1.3	38.1 ± 3.7	74.3 ± 3.8	75.6 ± 2.1	77.9 ± 2.2	76.6 ± 2.8
FPT	84.8 ± 1.1	50.7 ± 3.0	87.3 ± 2.9	51.6 ± 3.6	94.0 ± 1.4	95.3 ± 0.9	94.7 ± 1.1	94.9 ± 1.1
Time-LLM	84.4 ± 1.8	50.2 ± 1.6	85.1 ± 2.6	50.1 ± 3.4	93.4 ± 1.2	94.2 ± 1.3	94.7 ± 1.0	94.4 ± 1.1
ViTST	85.1 ± 0.8	51.1 ± 4.1	89.2 ± 2.0	53.1 ± 3.4	95.8 ± 1.3	96.2 ± 1.3	96.1 ± 1.1	96.5 ± 1.2
GRU-D	81.9 ± 2.1	46.1 ± 4.7	83.9 ± 1.7	46.9 ± 2.1	83.3 ± 1.6	84.6 ± 1.2	85.2 ± 1.6	84.8 ± 1.2
SeFT	73.9 ± 2.5	31.1 ± 4.1	81.2 ± 2.3	41.9 ± 3.1	67.1 ± 2.2	70.0 ± 2.4	68.2 ± 1.5	68.5 ± 1.8
mTAND	84.2 ± 0.8	48.2 ± 3.4	84.4 ± 1.3	50.6 ± 2.0	92.9 ± 0.8	93.8 ± 0.8	94.0 ± 0.9	93.8 ± 0.8
Raindrop	82.8 ± 1.7	44.0 ± 3.0	87.0 ± 2.3	51.8 ± 5.5	88.5 ± 1.5	89.9 ± 1.5	89.9 ± 0.6	89.8 ± 1.0
Warpformer	86.6 ± 0.8	55.5 ± 3.5	88.1 ± 2.5	56.1 ± 4.4	95.1 ± 0.8	95.7 ± 0.8	95.7 ± 0.9	95.7 ± 0.8
Latent-ODE	86.9 ± 0.2	56.7 ± 1.2	88.7 ± 0.4	56.3 ± 1.2	95.0 ± 0.5	95.2 ± 0.3	94.9 ± 1.3	95.6 ± 0.6
t-PatchGNN	86.1 ± 1.1	56.4 ± 0.9	90.6 ± 1.3	57.3 ± 2.9	96.1 ± 0.4	96.4 ± 0.7	96.9 ± 0.3	96.4 ± 1.3
ISTS-PLM	**87.6** ± 1.4	**57.6** ± 3.3	89.4 ± 2.2	56.9 ± 5.0	96.3 ± 0.5	96.9 ± 1.0	96.8 ± 0.4	96.8 ± 0.7
TimeCHEAT	84.5 ± 0.7	48.2 ± 1.9	89.5 ± 1.9	56.1 ± 4.6	96.5 ± 0.6	97.1 ± 0.5	96.9 ± 0.6	97.0 ± 0.5
GraFITi	83.2 ± 0.8	52.1 ± 1.1	88.2 ± 0.1	58.3 ± 0.5	95.2 ± 0.9	96.8 ± 0.7	94.6 ± 0.3	94.7 ± 0.6
KAFNet	86.2 ± 0.4	52.1 ± 0.8	90.1 ± 0.3	59.3 ± 0.6	95.5 ± 0.6	97.1 ± 0.6	96.1 ± 0.4	95.5 ± 0.8
APN	85.7 ± 0.5	53.3 ± 0.9	89.9 ± 0.2	58.9 ± 0.7	95.6 ± 0.5	96.8 ± 0.5	95.9 ± 0.4	95.3 ± 0.4
TFMixer	86.3 ± 0.6	54.3 ± 0.7	89.6 ± 0.5	58.7 ± 0.8	95.8 ± 0.3	96.8 ± 0.6	96.1 ± 0.5	95.5 ± 0.5
HeteroNet	87.3 ± 0.3	54.1 ± 1.1	**91.8** ± 0.5	**62.2** ± 0.5	**97.0** ± 0.1	**97.8** ± 0.1	**97.3** ± 0.1	**97.5** ± 0.1

USHCN extend this pattern beyond clinical data. The consistent gains in both error metrics support its effectiveness for within-window value reconstruction under the evaluated sampling conditions.

Table 8: Complete interpolation results (mean ± std). Best and second-best results over all methods are shown in **red** and blue, respectively. Lower values are better. We use results reported in (Zhang et al., 2025) where available and reproduce the remaining baselines.

Method	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}
Trans-mean	7.76 ± 0.18	4.92 ± 0.07	1.87 ± 0.06	7.63 ± 0.18	3.37 ± 0.19	3.89 ± 0.11	7.73 ± 0.12	4.85 ± 0.08
Neural Flow	6.77 ± 0.06	4.60 ± 0.07	2.18 ± 0.11	8.30 ± 0.28	3.73 ± 0.06	4.20 ± 0.05	7.41 ± 0.09	4.82 ± 0.06
CRU	10.30 ± 0.10	5.47 ± 0.05	2.52 ± 0.04	8.87 ± 0.03	7.17 ± 0.32	6.35 ± 0.14	7.45 ± 0.11	4.82 ± 0.07
GRU-D	6.18 ± 0.23	4.35 ± 0.06	2.06 ± 0.05	7.83 ± 0.14	2.74 ± 0.09	3.40 ± 0.08	7.20 ± 0.06	4.42 ± 0.05
SeFT	9.46 ± 0.12	5.51 ± 0.12	2.12 ± 0.02	8.08 ± 0.11	14.95 ± 0.03	9.42 ± 0.02	6.77 ± 0.10	4.19 ± 0.06
mTAND	5.65 ± 0.08	4.18 ± 0.07	1.93 ± 0.05	7.49 ± 0.08	2.07 ± 0.17	2.74 ± 0.23	6.42 ± 0.08	4.33 ± 0.12
Raindrop	10.65 ± 0.12	5.80 ± 0.07	2.31 ± 0.04	8.62 ± 0.10	15.21 ± 0.12	9.51 ± 0.05	6.88 ± 0.08	3.65 ± 0.06
Warpformer	6.37 ± 0.34	4.43 ± 0.20	1.93 ± 0.06	7.57 ± 0.12	2.59 ± 0.15	3.20 ± 0.11	6.12 ± 0.06	3.38 ± 0.05
Latent-ODE	6.84 ± 0.34	4.67 ± 0.14	1.89 ± 0.08	7.67 ± 0.47	3.12 ± 0.22	3.76 ± 0.16	6.22 ± 0.04	3.51 ± 0.09
t-PatchGNN	4.75 ± 0.03	3.59 ± 0.03	1.55 ± 0.08	6.25 ± 0.21	1.95 ± 0.12	2.54 ± 0.14	5.63 ± 0.04	3.25 ± 0.03
ISTS-PLM	4.55 ± 0.08	3.37 ± 0.02	1.47 ± 0.01	5.94 ± 0.01	1.93 ± 0.01	2.52 ± 0.01	5.45 ± 0.08	2.92 ± 0.06
TimeCHEAT	5.18 ± 0.08	3.55 ± 0.06	1.42 ± 0.08	5.91 ± 0.05	2.12 ± 0.11	2.89 ± 0.04	5.46 ± 0.07	2.95 ± 0.06
GraFITi	5.11 ± 0.13	3.61 ± 0.07	1.44 ± 0.10	5.94 ± 0.06	2.43 ± 0.06	3.12 ± 0.05	6.55 ± 0.11	3.01 ± 0.09
KAFNet	6.52 ± 0.03	4.26 ± 0.05	1.58 ± 0.08	6.12 ± 0.11	2.93 ± 0.03	3.37 ± 0.08	7.68 ± 0.06	3.05 ± 0.04
APN	6.12 ± 0.04	4.04 ± 0.08	1.66 ± 0.05	6.02 ± 0.03	1.93 ± 0.05	2.89 ± 0.06	7.78 ± 0.03	3.29 ± 0.02
TFMixer	5.53 ± 0.04	4.04 ± 0.06	1.88 ± 0.07	5.92 ± 0.06	2.24 ± 0.05	3.05 ± 0.02	7.06 ± 0.08	3.49 ± 0.05
HeteroNet	**4.31** ± 0.03	**3.13** ± 0.03	**1.12** ± 0.04	**5.38** ± 0.11	**1.76** ± 0.01	**2.28** ± 0.02	**5.06** ± 0.01	**2.77** ± 0.01

E.3 Extrapolation

Tab. 9 shows that HeteroNet leads five of eight metrics: both MSE and MAE on PhysioNet and Human Activity, and MAE on USHCN. On MIMIC, KAFNet (Zhou et al., 2026c) achieves lower MSE and MAE; on USHCN, TimeCHEAT (Liu et al., 2025a) achieves lower MSE. Compared with the interpolation results, these differences on MIMIC and USHCN show that strong within-window reconstruction does not guarantee the lowest future-prediction error on every dataset.

E.4 Temporal Point Process

Tab. 10 shows that HeteroNet has the lowest event-time RMSE on Amazon, Retweet and Taxi. It also has the lowest event-type error on Taobao. The relative strengths differ between prediction targets: on

Table 9: Complete extrapolation results (mean ± std). Best and second-best results over all methods are shown in **red** and blue, respectively. Lower values are better. For those baselines that already included in (Zhang et al., 2025), we use the results, and reproduced others.

Method	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}
Trans-mean	7.45 ± 0.26	4.97 ± 0.16	1.76 ± 0.02	7.59 ± 0.10	3.65 ± 0.11	4.20 ± 0.11	6.11 ± 0.15	3.87 ± 0.05
Neural Flow	7.20 ± 0.07	4.67 ± 0.04	1.87 ± 0.05	8.03 ± 0.19	4.05 ± 0.13	4.46 ± 0.09	5.35 ± 0.04	3.24 ± 0.06
CRU	8.56 ± 0.26	5.16 ± 0.09	1.97 ± 0.02	7.93 ± 0.19	6.97 ± 0.78	6.30 ± 0.47	6.07 ± 0.06	3.53 ± 0.13
GRU-D	5.59 ± 0.09	4.08 ± 0.05	1.76 ± 0.03	7.53 ± 0.09	2.94 ± 0.05	3.53 ± 0.06	5.54 ± 0.13	3.42 ± 0.20
SeFT	9.22 ± 0.18	5.40 ± 0.08	1.87 ± 0.01	7.84 ± 0.08	12.20 ± 0.17	8.43 ± 0.07	5.83 ± 0.10	3.71 ± 0.09
mTAND	6.23 ± 0.24	4.51 ± 0.17	1.85 ± 0.06	7.73 ± 0.13	3.22 ± 0.07	3.81 ± 0.07	5.34 ± 0.04	3.26 ± 0.09
Raindrop	9.82 ± 0.08	5.57 ± 0.06	1.99 ± 0.03	8.27 ± 0.07	14.92 ± 0.14	9.45 ± 0.05	5.79 ± 0.03	3.68 ± 0.12
Warpformer	5.94 ± 0.35	4.21 ± 0.12	1.73 ± 0.04	7.58 ± 0.13	2.79 ± 0.04	3.39 ± 0.03	5.25 ± 0.04	3.23 ± 0.02
Latent-ODE	6.05 ± 0.57	4.23 ± 0.17	1.89 ± 0.19	8.11 ± 0.52	3.34 ± 0.11	3.94 ± 0.12	5.62 ± 0.03	3.62 ± 0.10
t-PatchGNN	4.98 ± 0.08	3.72 ± 0.03	1.69 ± 0.03	7.22 ± 0.09	2.66 ± 0.03	3.15 ± 0.02	5.00 ± 0.04	3.10 ± 0.04
ISTS-PLM	4.92 ± 0.05	3.65 ± 0.04	1.64 ± 0.02	7.02 ± 0.14	2.58 ± 0.03	3.12 ± 0.04	5.06 ± 0.03	2.88 ± 0.02
TimeCHEAT	5.07 ± 0.02	3.90 ± 0.01	1.70 ± 0.02	7.43 ± 0.06	4.05 ± 0.06	4.65 ± 0.03	**4.45** ± 0.03	3.12 ± 0.05
GraFITi	6.01 ± 0.06	3.73 ± 0.03	1.72 ± 0.03	6.95 ± 0.05	2.73 ± 0.03	3.15 ± 0.02	5.11 ± 0.02	3.09 ± 0.04
KAFNet	5.90 ± 0.02	3.54 ± 0.01	**1.59** ± 0.02	**6.84** ± 0.06	2.55 ± 0.05	3.07 ± 0.05	5.02 ± 0.04	2.98 ± 0.02
APN	5.04 ± 0.02	3.63 ± 0.04	1.85 ± 0.02	7.01 ± 0.04	2.63 ± 0.06	3.15 ± 0.04	5.13 ± 0.03	3.04 ± 0.04
TFMixer	5.01 ± 0.03	3.62 ± 0.05	1.88 ± 0.04	7.03 ± 0.06	2.60 ± 0.04	3.14 ± 0.02	5.11 ± 0.01	3.00 ± 0.06
HeteroNet	**4.72** ± 0.02	**3.46** ± 0.02	1.68 ± 0.06	6.98 ± 0.08	**2.53** ± 0.01	**3.04** ± 0.01	5.04 ± 0.01	**2.83** ± 0.02

Taxi, NHP (Mei & Eisner, 2017) has lower type error, while on Taobao, AttNHP and FullyNN (Omi et al., 2019) have lower time RMSE. In this task, event values are fixed at one, and the TPP head uses the token representation without Kernel Fusion. The component ablation in Fig. 5 evaluates this configuration on Amazon event-time prediction.

Table 10: Complete TPP event-prediction results (mean ± std). Best and second-best results over all methods are shown in **red** and blue, respectively. Lower values are better; unavailable results are denoted by "–". Results for dedicated TPP baselines are taken from EasyTPP (Xue et al., 2024); we reproduce the remaining baselines.

Method	Amazon		Retweet		Taxi		Taobao	
	RMSE	Err(%)	RMSE	Err(%)	RMSE	Err(%)	RMSE	Err(%)
MHP	0.635 ± 0.005	75.9 ± 0.50	22.92 ± 0.212	55.7 ± 0.40	0.382 ± 0.002	9.5 ± 0.04	0.539 ± 0.004	68.1 ± 0.40
RMTPP	0.620 ± 0.005	68.1 ± 0.60	22.31 ± 0.209	44.1 ± 0.30	0.371 ± 0.003	9.5 ± 0.03	0.531 ± 0.005	55.8 ± 0.40
NHP	0.621 ± 0.005	67.1 ± 0.60	21.90 ± 0.184	40.0 ± 0.20	0.369 ± 0.003	**8.5** ± 0.05	0.531 ± 0.005	54.2 ± 0.60
SAHP	0.619 ± 0.005	67.7 ± 0.60	22.40 ± 0.301	41.6 ± 0.20	0.372 ± 0.003	9.8 ± 0.08	0.532 ± 0.004	54.6 ± 0.20
THP	0.621 ± 0.003	66.1 ± 0.70	22.01 ± 0.188	41.5 ± 0.30	0.370 ± 0.003	8.7 ± 0.06	0.531 ± 0.003	53.6 ± 0.40
AttNHP	0.621 ± 0.005	**65.3** ± 0.60	22.19 ± 0.180	40.1 ± 0.30	0.371 ± 0.003	8.7 ± 0.04	**0.529** ± 0.005	53.7 ± 0.10
ODETPP	0.620 ± 0.006	65.8 ± 0.80	22.48 ± 0.175	43.2 ± 0.40	0.371 ± 0.003	10.5 ± 0.08	0.533 ± 0.005	55.4 ± 0.70
FullyNN	0.615 ± 0.005	–	21.92 ± 0.159	–	0.373 ± 0.003	–	**0.529** ± 0.005	–
IFTPP	0.618 ± 0.005	67.5 ± 0.70	22.18 ± 0.204	**39.7** ± 0.30	0.377 ± 0.003	8.6 ± 0.60	0.531 ± 0.005	55.4 ± 0.40
t-PatchGNN	0.638 ± 0.013	74.5 ± 0.1	24.1 ± 0.2	43.3 ± 0.5	0.382 ± 0.040	12.1 ± 0.6	0.562 ± 0.012	72.4 ± 0.11
ISTS-PLM	0.646 ± 0.012	69.7 ± 0.1	24.4 ± 0.2	42.9 ± 0.6	0.384 ± 0.043	12.0 ± 0.7	0.545 ± 0.007	70.9 ± 0.08
TimeCHEAT	0.641 ± 0.011	70.9 ± 0.2	24.0 ± 0.2	45.6 ± 0.4	0.380 ± 0.041	12.2 ± 0.4	0.555 ± 0.010	69.3 ± 0.22
GraFITi	0.639 ± 0.013	69.6 ± 0.2	24.4 ± 0.2	42.5 ± 0.3	0.386 ± 0.042	11.8 ± 0.7	0.561 ± 0.012	70.2 ± 0.07
KAFNet	0.555 ± 0.022	68.8 ± 0.1	24.0 ± 0.2	43.2 ± 0.4	0.382 ± 0.041	12.2 ± 0.6	0.558 ± 0.005	71.3 ± 0.33
APN	0.655 ± 0.018	71.8 ± 0.1	23.9 ± 0.2	42.8 ± 0.9	0.388 ± 0.043	11.9 ± 0.7	0.556 ± 0.006	73.3 ± 0.23
TFMixer	0.619 ± 0.012	67.9 ± 0.28	22.4 ± 0.171	41.7 ± 0.2	0.373 ± 0.036	9.8 ± 0.10	0.565 ± 0.013	56.1 ± 0.12
HeteroNet	**0.514** ± 0.004	66.7 ± 0.26	**21.65** ± 0.1	40.6 ± 0.2	**0.368** ± 0.002	10.6 ± 0.4	0.552 ± 0.008	**41.9** ± 0.33

E.5 Generation

The three generation tables compare dedicated generative models and general-purpose IMTS baselines at missing rates of 30%, 50%, and 70%. Results for the general-purpose IMTS baselines are obtained from our reproductions. Results for the dedicated generative models are taken from Diff-MN (Zhang et al., 2026b), which does not report across-run standard deviations for these comparisons. To present all methods consistently, we therefore report point estimates for every method, including HeteroNet.

F Limitations

While HeteroNet shows strong overall results across five IMTS analysis tasks and 15 datasets, two aspects warrant further exploration. First, the experiments focus on widely used public benchmarks;

Table 11: Complete generation results for sequence length 12. Best and second-best results are highlighted in **red** and blue, respectively. Lower values are better. Results for dedicated generation baselines are taken from Diff-MN (Zhang et al., 2026b); we reproduce the remaining baselines.

Metric	Method	30% Dropped				50% Dropped				70% Dropped			
		Sines	Stocks	Energy	MuJoCo	Sines	Stocks	Energy	MuJoCo	Sines	Stocks	Energy	MuJoCo
DS ($\times 10^{-2}$)	KoVAE	21.8	10.3	50.0	35.8	20.2	10.6	50.0	39.0	21.1	22.8	50.0	45.1
	GT-GAN	30.0	38.5	49.9	49.8	42.3	35.7	49.9	49.6	43.4	37.0	49.8	49.7
	TimeGAN-NCDE	46.8	36.9	49.8	48.7	42.2	47.2	49.9	49.9	42.9	47.9	50.0	49.8
	TimeVAE-NCDE	35.4	49.3	49.6	43.3	46.6	49.7	49.1	47.7	50.0	49.7	49.8	48.0
	Diffusion-NCDE	26.4	42.8	42.6	32.3	35.3	47.2	48.2	43.7	48.0	49.1	49.7	47.5
	ProFITi	29.0	44.0	49.2	49.0	33.9	47.8	49.8	49.6	41.0	48.6	50.0	49.7
	HeTVAE	50.0	36.8	49.3	49.9	50.0	47.8	49.9	50.0	50.0	49.3	50.0	50.0
	Diff-MN	7.7	**8.4**	38.6	26.8	13.0	**4.2**	47.2	41.7	22.2	5.4	49.4	45.0
	t-PatchGNN	32.2	16.9	21.4	46.8	19.5	15.6	32.5	28.5	19.2	21.1	46.2	45.2
	ISTS-PLM	28.4	18.7	33.9	45.2	22.1	17.5	31.8	26.9	8.2	23.7	48.3	46.6
	TimeCHEAT	34.1	15.8	25.6	47.9	18.3	20.5	35.2	29.1	21.8	24.4	44.7	9.3
	GraFITi	30.7	20.2	28.1	43.5	17.9	16.4	33.6	27.4	19.7	46.2	47.1	6.5
	KAFNet	29.5	17.3	24.2	49.1	20.6	14.8	32.1	25.8	18.9	25.3	48.8	7.1
	APN	31.8	19.4	26.7	44.6	21.3	15.2	34.9	28.7	22.4	27.8	46.5	8.8
	TFMixer	27.6	16.1	30.4	46.3	18.8	17.9	30.2	24.5	24.6	24.9	43.8	5.7
	HeteroNet	**4.8**	13.7	**19.0**	**23.3**	**12.2**	17.7	**22.4**	**22.6**	**1.5**	**2.8**	**32.2**	**4.1**
MDD ($\times 10^{-1}$)	KoVAE	101.4	2.9	3.4	3.3	101.2	2.9	3.8	3.7	101.3	3.1	4.0	6.4
	GT-GAN	35.7	7.5	6.0	5.4	39.0	6.0	4.1	5.7	33.8	7.3	5.6	5.9
	TimeGAN-NCDE	46.0	11.1	10.2	12.6	48.0	16.3	14.0	14.9	53.5	17.2	15.0	17.6
	TimeVAE-NCDE	86.6	7.1	4.5	4.6	60.5	8.6	5.9	5.9	51.8	10.5	6.5	7.1
	Diffusion-NCDE	21.5	3.3	3.1	2.8	21.3	5.4	3.7	4.3	29.0	8.5	5.1	5.1
	ProFITi	12.8	3.2	2.5	3.0	13.4	4.2	3.4	3.6	15.8	4.4	4.4	4.5
	HeTVAE	35.9	6.0	6.4	5.1	42.5	7.5	11.1	10.9	51.3	9.9	12.7	11.1
	Diff-MN	10.2	1.6	3.0	3.0	13.4	2.0	2.5	3.2	17.8	2.2	3.2	3.4
	t-PatchGNN	44.2	12.3	9.8	13.5	50.1	15.7	13.2	15.1	55.2	16.8	14.5	18.3
	ISTS-PLM	47.8	10.5	11.4	12.1	45.6	17.1	15.3	16.2	52.1	18.4	13.8	8.9
	TimeCHEAT	43.5	13.2	10.7	14.2	49.3	16.5	14.1	15.8	54.7	17.9	15.2	19.1
	GraFITi	48.9	11.8	9.5	13.1	46.4	15.2	13.6	16.5	51.8	18.7	14.3	7.2
	KAFNet	45.1	12.7	11.6	12.4	52.0	17.3	15.8	14.7	56.1	16.1	13.9	20.5
	APN	42.8	10.9	10.1	13.8	47.5	16.8	14.4	17.2	53.9	17.5	15.6	8.3
	TFMixer	49.6	13.1	12.3	11.9	44.7	15.9	13.7	15.3	50.2	18.2	14.8	21.2
	HeteroNet	**8.2**	**1.3**	**2.1**	**1.9**	**5.9**	**1.6**	**1.6**	**2.3**	**5.6**	**1.8**	**1.8**	**2.4**
KL ($\times 10^{-2}$)	KoVAE	994.1	13.9	7.6	16.8	951.1	14.9	8.4	28.2	978.8	15.1	11.1	57.9
	GT-GAN	14.4	28.3	6.3	2.0	18.7	21.2	5.0	2.9	16.4	31.4	6.6	3.1
	TimeGAN-NCDE	523.0	174.2	21.0	57.5	112.5	740.1	71.6	99.7	1517.6	1214.8	85.4	239.7
	TimeVAE-NCDE	355.1	29.8	8.9	15.8	277.6	34.5	12.8	24.5	211.3	53.4	14.8	33.8
	Diffusion-NCDE	0.4	**1.2**	**0.4**	0.7	2.8	4.0	1.9	3.6	7.4	9.5	4.6	4.2
	ProFITi	7.8	8.4	2.2	1.6	8.2	12.4	3.2	1.6	11.0	13.7	4.4	3.2
	HeTVAE	3.6	30.2	37.6	10.2	265.5	46.0	112.4	93.4	28.8	78.0	127.5	107.9
	Diff-MN	0.7	5.0	1.9	1.7	1.4	7.5	2.5	1.8	3.3	6.6	1.3	2.4
	t-PatchGNN	1.2	3.1	1.5	0.8	2.4	8.2	2.1	0.9	6.3	9.1	1.8	1.6
	ISTS-PLM	9.3	12.1	3.7	2.8	11.5	16.2	4.9	2.5	14.8	17.3	5.6	4.2
	TimeCHEAT	6.2	7.4	2.2	1.4	7.8	10.9	3.2	1.6	9.2	11.8	3.9	2.7
	GraFITi	3.1	4.2	1.8	0.9	4.5	6.7	2.3	1.2	5.8	7.9	2.6	1.9
	KAFNet	10.8	13.6	4.1	3.2	12.9	17.5	5.2	2.9	15.7	18.4	5.9	4.6
	APN	2.1	2.5	1.1	0.7	1.9	5.3	1.6	0.8	4.2	**6.1**	1.4	1.2
	TFMixer	8.7	11.2	3.4	2.4	10.1	14.8	4.5	2.3	13.2	16.1	5.1	3.8
	HeteroNet	**0.3**	1.4	0.6	**0.4**	**1.2**	**3.2**	**1.4**	**0.3**	**2.2**	9.5	**1.2**	**0.6**

evaluation in additional application domains and under more diverse sampling conditions would further assess its general applicability. Second, the tasks share Observation Encoding, Sparse Pattern Tokenization, and Pattern Interaction, but each model is trained separately with a task-specific output head. TPP additionally omits Kernel Fusion. Joint multi-task pretraining may improve knowledge sharing and adaptation across IMTS tasks.

Table 12: Complete generation results for sequence length 24. Best and second-best results are highlighted in **red** and blue, respectively. Lower values are better. Results for dedicated generation baselines are taken from Diff-MN (Zhang et al., 2026b); we reproduce the remaining baselines.

Metric	Method	30% Dropped				50% Dropped				70% Dropped			
		Sines	Stocks	Energy	MuJoCo	Sines	Stocks	Energy	MuJoCo	Sines	Stocks	Energy	MuJoCo
DS ($\times 10^{-2}$)	KoVAE	5.5	28.5	50.0	35.8	9.0	20.8	50.0	36.3	21.2	23.6	50.0	40.3
	GT-GAN	27.6	30.5	50.0	48.0	33.8	32.5	50.0	49.1	28.6	27.3	50.0	48.9
	TimeGAN-NCDE	45.7	35.1	50.0	48.6	43.3	45.4	50.0	49.7	43.8	47.2	49.9	49.9
	TimeVAE-NCDE	19.1	48.1	49.9	32.4	17.5	49.3	49.9	46.2	43.1	49.0	49.9	46.4
	Diffusion-NCDE	14.8	46.0	45.5	37.3	28.6	49.0	48.9	40.9	39.2	48.9	49.6	47.0
	ProFITi	29.8	45.9	49.8	49.6	36.5	48.7	49.9	49.8	40.9	49.2	50.0	49.8
	HeTVAE	48.0	30.6	49.8	49.9	50.0	40.1	50.0	49.9	50.0	47.6	50.0	50.0
	Diff-MN	6.9	17.1	41.4	24.9	7.2	15.1	49.1	**30.6**	12.7	**17.9**	49.9	**34.7**
	t-PatchGNN	31.2	47.3	48.1	46.8	38.7	49.5	48.3	47.2	42.1	48.9	49.4	48.1
	ISTS-PLM	26.5	44.2	47.6	48.9	33.4	46.8	49.6	43.4	37.8	47.1	49.7	48.5
	TimeCHEAT	35.9	49.1	47.4	48.2	41.3	45.6	48.9	49.1	44.6	48.4	49.8	47.9
	GraFITi	28.3	43.7	49.2	47.5	34.2	48.1	49.4	48.7	39.5	49.6	48.8	49.2
	KAFNet	37.6	48.5	48.7	49.3	43.1	49.8	49.2	48.9	45.2	47.9	49.5	48.6
	APN	25.8	46.1	49.5	48.4	32.9	47.3	48.8	49.6	38.2	48.7	49.9	48.2
	TFMixer	33.7	44.4	48.9	49.1	39.8	49.4	49.7	48.3	43.5	48.1	49.6	49.4
	HeteroNet	**3.3**	**13.4**	**37.2**	**21.2**	**5.3**	**12.4**	**41.8**	32.3	**11.7**	24.1	**39.6**	40.2
MDD ($\times 10^{-1}$)	KoVAE	27.6	9.1	3.5	3.4	33.1	8.3	3.6	3.8	52.2	7.6	3.7	6.3
	GT-GAN	22.6	5.5	4.3	5.8	24.7	5.6	4.0	5.3	22.8	5.3	4.1	5.6
	TimeGAN-NCDE	31.6	11.3	10.5	11.7	35.6	16.0	12.3	15.5	36.1	15.9	16.1	16.5
	TimeVAE-NCDE	34.1	5.8	4.2	3.1	30.4	7.1	5.2	4.1	28.6	8.0	6.0	5.3
	Diffusion-NCDE	13.0	4.0	2.5	3.4	14.0	4.0	3.6	3.4	16.7	6.9	5.6	4.9
	ProFITi	9.6	3.1	2.6	2.8	14.5	4.7	3.6	3.3	16.5	4.4	4.4	4.1
	HeTVAE	27.8	5.8	6.1	7.2	34.3	7.9	9.9	10.9	36.8	10.7	12.3	14.5
	Diff-MN	7.5	2.7	2.2	3.4	7.8	2.5	2.2	3.4	9.7	3.0	2.7	2.9
	t-PatchGNN	11.2	3.8	2.4	3.1	12.8	5.2	3.8	2.9	17.3	4.6	4.2	3.7
	ISTS-PLM	7.4	2.9	3.2	2.5	16.1	4.1	4.3	3.5	13.2	5.1	3.9	4.6
	TimeCHEAT	13.7	4.5	2.7	3.4	10.2	3.9	4.1	2.8	18.4	4.3	3.5	4.0
	GraFITi	8.1	3.3	3.8	2.6	15.6	4.8	3.4	3.8	11.9	4.9	4.5	3.2
	KAFNet	14.3	4.2	2.5	3.2	9.8	4.4	3.9	2.7	16.8	3.7	4.1	4.3
	APN	6.9	3.6	3.1	2.9	17.2	5.0	4.0	3.6	12.5	4.2	3.8	3.9
	TFMixer	10.5	4.1	2.8	3.3	13.9	4.3	3.7	3.1	15.1	4.8	4.3	4.2
	HeteroNet	**4.5**	**1.5**	**1.4**	**2.0**	**5.8**	**1.8**	**1.8**	**2.3**	**5.3**	**2.4**	**2.2**	**2.6**
KL ($\times 10^{-2}$)	KoVAE	169.6	134.7	5.8	15.0	186.1	120.5	6.1	19.8	284.0	138.0	7.1	50.8
	GT-GAN	6.5	16.3	4.2	3.2	13.9	16.1	5.3	1.9	19.0	18.4	5.1	3.4
	TimeGAN-NCDE	457.3	138.0	22.7	66.6	433.9	588.7	48.9	124.8	485.4	1050.8	111.9	140.9
	TimeVAE-NCDE	218.0	20.9	6.7	4.6	129.1	24.5	10.2	7.0	103.8	28.4	13.2	11.2
	Diffusion-NCDE	6.0	43.8	7.3	3.3	5.3	44.0	9.9	4.1	7.7	49.8	11.4	4.7
	ProFITi	12.8	9.1	2.8	1.0	13.7	16.1	3.6	1.2	18.9	12.3	4.3	2.6
	HeTVAE	32.3	79.2	17.5	31.8	871.2	120.3	49.5	91.5	33.7	139.8	142.9	234.3
	Diff-MN	1.1	8.4	**0.9**	0.7	2.2	9.2	1.4	0.6	3.5	10.5	2.3	1.3
	t-PatchGNN	11.3	8.6	3.1	1.3	14.2	15.4	3.9	1.1	17.6	11.8	4.1	2.9
	ISTS-PLM	15.2	10.7	2.5	1.8	12.1	17.3	3.4	1.6	19.4	13.5	4.7	2.2
	TimeCHEAT	9.7	7.4	3.4	0.9	16.8	14.2	4.1	1.4	16.1	10.2	3.8	3.2
	GraFITi	13.1	11.2	2.9	1.5	11.6	18.7	3.7	1.9	20.1	12.9	4.5	2.4
	KAFNet	10.4	8.9	3.5	1.1	15.3	13.8	4.2	1.3	17.2	11.5	3.9	3.1
	APN	14.6	12.3	2.6	1.7	13.4	16.5	3.5	1.8	18.3	14.1	4.6	2.7
	TFMixer	8.8	7.2	3.7	0.8	17.1	11.9	4.3	1.2	15.7	**9.6**	4.0	3.3
	HeteroNet	**1.0**	**5.5**	1.2	**0.4**	**1.6**	**7.2**	**1.2**	**0.4**	**2.9**	10.4	**1.9**	**0.8**

Table 13: Complete generation results for sequence length 36. Best and second-best results are highlighted in **red** and blue, respectively. Lower values are better. Results for dedicated generation baselines are taken from Diff-MN (Zhang et al., 2026b); we reproduce the remaining baselines.

Metric	Method	30% Dropped				50% Dropped				70% Dropped			
		Sines	Stocks	Energy	MuJoCo	Sines	Stocks	Energy	MuJoCo	Sines	Stocks	Energy	MuJoCo
DS ($\times 10^{-2}$)	KoVAE	15.2	28.6	42.9	39.9	22.2	24.7	50.0	43.9	26.1	17.6	50.0	42.2
	GT-GAN	33.1	27.2	49.9	48.9	48.7	29.7	50.0	49.4	30.3	23.4	50.0	48.9
	TimeGAN-NCDE	44.0	37.6	49.9	49.6	42.5	39.9	49.8	50.0	45.3	48.0	50.0	49.9
	TimeVAE-NCDE	15.7	43.7	50.0	36.8	28.2	46.4	49.9	45.1	34.7	49.0	49.8	48.5
	Diffusion-NCDE	18.9	43.5	49.8	36.8	35.2	46.2	49.6	43.6	39.1	49.3	49.7	47.4
	ProFITi	34.2	48.1	49.9	49.8	35.4	49.3	50.0	49.8	35.7	49.4	50.0	49.8
	HeTVAE	49.4	30.9	49.9	49.8	50.0	44.9	50.0	49.9	50.0	49.3	50.0	50.0
	Diff-MN	16.9	17.0	46.6	36.3	18.3	21.8	49.9	40.3	19.6	**8.5**	49.9	38.1
	t-PatchGNN	18.3	42.1	49.7	38.2	25.6	47.8	49.3	44.2	32.1	48.4	48.9	47.3
	ISTS-PLM	22.9	45.8	48.9	35.4	31.7	44.6	49.6	46.9	38.5	47.2	49.4	49.1
	TimeCHEAT	12.1	41.3	47.1	40.7	27.4	48.9	48.7	43.5	36.2	49.5	49.9	46.8
	GraFITi	19.8	44.2	49.1	37.1	29.5	45.3	50.0	47.6	35.3	48.1	49.7	48.2
	KAFNet	26.3	46.5	49.4	34.2	23.7	47.1	49.8	44.8	39.1	49.3	49.5	47.9
	APN	14.2	40.7	48.8	42.3	30.8	46.2	49.2	45.9	33.6	48.6	48.6	49.0
	TFMixer	21.4	47.3	49.9	36.5	25.1	49.1	49.6	43.2	37.8	47.9	49.3	48.7
	HeteroNet	**7.7**	**15.6**	**42.6**	**31.4**	**12.5**	**15.8**	**42.1**	**32.7**	**12.1**	12.3	**31.5**	**26.4**
MDD ($\times 10^{-1}$)	KoVAE	25.1	9.3	3.5	4.6	33.2	8.7	3.7	5.1	38.3	8.0	3.6	5.9
	GT-GAN	21.0	5.9	6.7	5.6	25.7	6.0	6.1	6.9	20.2	5.4	6.2	7.4
	TimeGAN-NCDE	26.7	10.9	10.2	11.1	28.6	15.9	14.1	15.7	29.8	15.8	15.7	17.0
	TimeVAE-NCDE	22.3	5.0	4.6	2.4	22.0	6.0	4.9	3.3	21.4	6.7	5.6	4.6
	Diffusion-NCDE	9.3	3.0	3.2	3.1	9.6	4.9	3.3	3.5	14.2	6.8	4.8	4.7
	ProFITi	9.4	2.5	2.6	2.9	10.2	3.9	3.7	3.4	11.6	4.5	4.4	4.5
	HeTVAE	31.0	6.2	6.3	7.3	26.7	7.8	11.8	11.0	30.9	10.4	14.7	16.4
	Diff-MN	10.9	3.2	2.9	4.0	11.7	3.9	2.9	2.9	11.7	3.8	**2.5**	**2.7**
	t-PatchGNN	10.1	3.7	2.9	3.3	8.4	5.2	3.6	3.2	13.5	6.1	4.5	4.9
	ISTS-PLM	7.2	2.8	3.5	3.0	11.3	4.5	3.1	3.8	12.0	7.2	5.1	4.3
	TimeCHEAT	12.5	4.3	3.0	3.4	8.7	5.6	3.9	3.3	15.1	5.9	4.2	4.6
	GraFITi	8.6	3.2	3.6	2.9	10.4	4.7	3.2	3.7	11.8	6.5	4.9	4.2
	KAFNet	11.2	4.1	2.7	3.2	9.1	5.3	3.7	3.4	14.7	6.3	4.4	5.0
	APN	6.8	2.9	3.4	3.1	12.1	4.4	3.3	3.6	10.9	7.4	5.2	4.1
	TFMixer	13.4	4.2	3.1	3.3	7.9	5.8	3.8	3.2	13.1	6.2	4.6	4.8
	HeteroNet	**2.9**	**1.5**	**1.3**	**2.2**	**5.6**	**2.4**	**2.4**	**2.6**	**6.6**	**2.9**	**2.5**	3.3
KL ($\times 10^{-2}$)	KoVAE	87.8	155.5	6.1	7.2	278.5	122.4	5.7	11.5	393.5	107.2	5.7	18.8
	GT-GAN	8.7	16.3	6.4	1.2	13.4	17.1	5.1	3.5	13.4	17.1	5.4	7.8
	TimeGAN-NCDE	508.6	210.4	34.1	44.0	339.8	703.9	55.1	121.4	1740.3	1223.5	101.2	162.5
	TimeVAE-NCDE	26.5	16.6	10.7	2.1	39.5	18.6	7.4	3.8	43.7	21.6	12.8	6.7
	Diffusion-NCDE	8.7	23.5	15.1	4.2	8.3	33.4	4.4	7.1	12.3	39.8	16.3	3.9
	ProFITi	8.0	5.4	2.2	1.2	17.2	11.5	3.6	1.1	19.2	12.1	4.7	2.9
	HeTVAE	317.4	216.0	19.4	35.3	134.0	253.7	113.1	114.0	683.8	265.9	203.0	318.4
	Diff-MN	2.0	8.9	3.1	3.9	3.3	9.4	2.2	0.9	3.1	9.1	1.7	1.4
	t-PatchGNN	1.8	8.2	3.4	4.1	3.6	10.1	2.5	1.2	2.9	8.7	1.9	1.6
	ISTS-PLM	3.2	9.6	2.8	3.7	2.1	**8.5**	3.1	1.1	3.5	9.3	2.2	0.8
	TimeCHEAT	2.7	7.8	3.6	4.3	3.9	9.8	1.9	1.5	2.3	**8.3**	2.4	1.8
	GraFITi	1.5	9.2	3.0	3.5	2.8	9.7	2.6	0.7	3.7	8.9	**1.5**	2.1
	KAFNet	3.8	8.6	3.3	4.2	3.4	8.8	2.3	1.3	2.6	9.4	2.0	1.3
	APN	2.2	10.3	2.9	3.6	2.5	9.1	3.2	0.9	3.2	8.4	1.8	1.9
	TFMixer	3.5	7.9	3.7	4.0	3.7	9.5	2.0	1.4	2.8	9.0	2.3	1.5
	HeteroNet	**0.7**	**5.2**	**1.4**	**0.9**	**1.8**	10.2	**1.5**	**0.5**	**1.8**	15.4	2.1	**0.6**