CONDITIONAL DIFFUSION MODELS AS SELF-SUPERVISED LEARNING BACKBONE FOR IRREGULAR TIME SERIES

Hamed Shirzad *
University of British Columbia & Borealis AI
shirzad@cs.ubc.ca

Ruizhi Deng & He Zhao & Fred Tung
Borealis AI
{ruizhi.deng,he.zhao,frederick.tung}@borealisai.com

ABSTRACT

Irregular time series are ubiquitous in healthcare, with applications ranging from predicting patient health conditions to imputing missing values. Recent developments in conditional diffusion models, which predict missing values based on observed data, have shown significant promise for imputing regular time series. It also generalizes the self-supervised learning task of maskout reconstruction by replacing partial masking with injecting noise of variable scales to data and shows competitive results on image recognition. Despite the growing interest in diffusion models, their potential for irregular time series data, particularly in downstream tasks, remains underexplored. We propose a conditional diffusion model designed as a self-supervised learning backbone for such data, integrating a learnable time embedding and a cross-dimensional attention mechanism to address the data's complex temporal dynamics. This model not only suits conditional generation tasks naturally but also acquires hidden states beneficial for discriminative tasks. Empirical evidence demonstrates our model's superiority in both imputation and classification tasks.

1 INTRODUCTION

Observation sequences on irregular time grids are an integral part of healthcare data, offering key insights for predicting future health conditions. Generative pre-training has been a standard approach in self-supervised learning for sequential data, based on the belief that it enhances models' comprehension of the data (He et al., 2022; Devlin et al., 2018; Dai et al., 2017). Recently, diffusion models have significantly advanced generative modeling across multiple data domains (Ho et al., 2020; Song et al., 2020; Zheng et al., 2023; Hatamizadeh et al., 2023; Tashiro et al., 2021; Lüdke et al., 2023; Vignac et al., 2022). Despite their significant success in generating high-quality samples across several domains, diffusion models have only recently been explored for time series data. Tashiro et al. demonstrated a conditional diffusion model for regular time series imputation (CSDI). In their framework, the model predicts unobserved values from pure noise based on the observed parts of a sequence through iterative denoising. The training process involves recovering parts of the time series data partially corrupted by multi-scale noise, conditioned on the remaining clean data.

Meanwhile, Wei et al. proposed a self-supervised learning approach for images, DiffMAE, with a formulation similar to CSDI by predicting masked patches conditioned on the observed ones using diffusion models. We argue that such conditional diffusion models are promising for self-supervised learning in irregular time series data, particularly for recognition and generation tasks, for two main reasons: First, the success of self-supervised learning relies on data augmentation to create multiple views of data samples to learn meaningful representations. This framework's strategy of injecting variable-scale noise into randomly selected portions of a time series not only diversifies the data views

*Work done during an internship at Borealis AI

but also expands upon the conventional self-supervised learning technique of maskout reconstruction in sequential data, as seen in (Devlin et al., 2018; Li et al., 2023; Dong et al., 2023), by conditioning the model to rebuild noised segments using clean observations, with the reconstruction difficulty controlled by the noise scale. Specifically, at the maximum noise level, disrupted observations become indistinguishable from pure noise, thus noise injection and masking similarly remove all information from the data. Second, many regression problems, like imputation, in time series data can be formulated as conditional generation problems. The strong generative capabilities of diffusion models naturally lend the models to these tasks.

Despite promising potentials, existing methods struggle with the irregular temporal nature of healthcare data, characterized by unevenly spaced timestamps and missing dimensions in observations. To address these challenges, we adopt a learnable embedding for timestamps and a cross-dimensional attention mechanism for addressing the unobserved dimensions, extending the conditional diffusion model to irregular time series. While our model shares a similar architecture with CSDI for time series data, it uniquely adapts to irregularities through learnable time embeddings and showcases the hidden states' utility in downstream recognition tasks. Contrary to DiffMAE's encoder-decoder framework, our method uses a single model to simultaneously process noised and clean values, effectively denoising disrupted data. Experimental results confirm our model's effectiveness in imputation and its ability to learn meaningful hidden states for recognition tasks.

2 PRELIMINARIES

CSDI, DiffMAE, and our work share a common probabilistic formulation of modeling the conditional distribution of unobserved data given the observed data through diffusion models. We rely on the generative capabilities of this formulation for imputation and the conditional multi-scale denoising as a pretraining task for representation learning. This section introduces the general formulation of such conditional diffusion models. For illustrative purposes, our introduction is based on denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020), one of the most approachable diffusion model formulations and can be easily generalized to more advanced models. Given a multi-dimensional data sample $\mathbf{x}$, we assume it can be split into two non-overlapping parts: $\mathbf{x}^{co}$ is the part of $\mathbf{x}$ which is observed and $\mathbf{x}^{ta}$ is the unobserved part which we are interested in predicting conditioned on $\mathbf{x}^{co}$. The conditional diffusion model formulation models the distribution of $p(\mathbf{x}^{ta}|\mathbf{x}^{co})$.

In the forward diffusion process, noise is injected to $\mathbf{x}^{ta}$ gradually across multiple steps according to the following equation:

$$q(\mathbf{x}_s^{ta}|\mathbf{x}_{s-1}^{ta}) := \mathcal{N}(\mathbf{x}_s^{ta};\ \sqrt{1-\beta_s}\mathbf{x}_{s-1}^{ta},\ \beta_s\mathbf{I}), \tag{1}$$

where $\mathbf{x}_0^{ta} = \mathbf{x}^{ta}$, s is the index of diffusion steps from 1 to S, I is an identity matrix with the same dimension as $\mathbf{x}^{ta}$, and the β_ss are hyper-parameters controlling the speed of noise injection. At step S, signals in $\mathbf{x}^{ta}$ will be completely replaced by noise. An important property of this forward process is that the marginal distribution of $\mathbf{x}_s^{ta}$ given $\mathbf{x}_0^{ta}$ is also a closed-form Gaussian denoted as

$$q(\mathbf{x}_s^{ta}|\mathbf{x}_0^{ta}) = \mathcal{N}(\mathbf{x}_s^{ta}; \sqrt{\alpha_s}\mathbf{x}_0^{ta}, (1-\alpha_s)\mathbf{I}), \tag{2}$$

with α_ss derived from β_ss, enabling us to skip the intermediate steps and directly sample $\mathbf{x}_s^{ta}$. The hyper-parameters, β_ss, are picked so that α_s is close to zero when the diffusion step is close to S and $\mathbf{x}_S^{ta}$ is indistinguishable from standard Gaussian noise. The model is trained with the objective of predicting the noise injected to $\mathbf{x}_s^{ta}$ conditioned on $\mathbf{x}^{co}$ by minimizing the following loss:

$$\mathcal{L}_{\text{simple}}(\theta) := \mathbb{E}_{\mathbf{x}_0,s,\boldsymbol{\epsilon}}\big[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\alpha_s}\mathbf{x}_0^{ta} + \sqrt{1-\alpha_s}\boldsymbol{\epsilon}, s, \mathbf{x}_0^{co})||^2\big], \tag{3}$$

where ϵ_θ is a learnable denoising network, and ϵ is the added noise. This objective is mathematically equivalent to recovering $\mathbf{x}_0^{ta}$. For conditional data generation, we run the conditional diffusion model in a reverse process by first sampling $\hat{\mathbf{x}}_S^{ta}$ from Gaussian noise and applying the denoising network iteratively to input with the following equation:

$$\hat{\mathbf{x}}_{s-1}^{ta} = \frac{1}{\sqrt{\bar{\alpha}_s}}\left[\hat{\mathbf{x}}_s^{ta} - \frac{1-\bar{\alpha}_s}{\sqrt{1-\alpha_s}}\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_s^{ta}, t, \mathbf{x}^{co})\right] + \sigma_s\mathbf{z},\ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

where $\bar{\alpha}_s = \alpha_s/\alpha_{s-1}$ and $\bar{\alpha}_1 = \alpha_1$.

3 APPROACH

Let $\{(\tau^i, \mathbf{x}^i, \mathbf{m}^i)\}_{i=1}^L$ be an irregular time series of K features where $\mathbf{x}^i \in \mathbb{R}^K$ is the vector of observations at timestamp τ_i and $\mathbf{m}^i \in \{0,1\}^K$ is a binary mask indicating the measured features, i.e. observed values, with one. All unmeasured features have value zero. Alternatively, we can represent the sequence of observations and masks more compactly using two matrices, $\boldsymbol{X} \in \mathbb{R}^{K \times L}$ and $\boldsymbol{M} \in \{0,1\}^{K \times L}$ where the ith columns of $\boldsymbol{X}$ and $\boldsymbol{M}$ are $\mathbf{x}^i$ and $\mathbf{m}^i$ respectively. For simplicity of illustration, the two sets of notations will be used interchangeably throughout the section.

We follow a training setting similar to CSDI (Tashiro et al., 2021) by randomly selecting a subset of the indices $\{(i,k)|\boldsymbol{M}[i,k]=1\}$ as the context set co and all the remaining observed indices as the target set ta. We define $\boldsymbol{X}^{ta}$ and $\boldsymbol{X}^{co}$ as the sets of actually measured values in $\boldsymbol{X}$, as indicated by the mask matrix $\boldsymbol{M}$, with indices in the context and target sets respectively. With slight abuse of notations, $\boldsymbol{X}_s^{ta}$, $\mathbf{x}_s^i$, and $\boldsymbol{X}_s$ are used to denote the corresponding values of $\boldsymbol{X}^{ta}$, $\mathbf{x}^i$, and $\boldsymbol{X}$ after s steps of diffusion applied to the target indices. The remainder of this section will present two important techniques that help us extend the conditional diffusion model formulation to irregular time series data, learnable irregular time embedding and cross-dimensional transformer. Additionally, it will explain the approaches to using hidden states of the model for downstream recognition tasks.

3.1 LEARNABLE EMBEDDING FOR IRREGULAR TIMESTAMPS

Transformer models for regular time series typically rely on positional encoding to encode the order of data. A more effective inductive bias for irregular time series can be introduced through learnable continuous time embeddings, capable of capturing both linear and periodic characteristics of time. These embeddings are also used for the mTAN model (Shukla & Marlin, 2021). Given a time stamp τ, $E(\tau) \in \mathbb{R}^{d_\tau}$, is a learnable irregular time encoding with size $d_\tau > 1$, and its ith dimension is defined as:

$$E(\tau)[i] = \begin{cases} w_0 \cdot \tau + b_0, & i = 0 \\ \sin\left(w_i \cdot \tau + b_i\right), & i \in \{1, 2, \ldots, d_\tau - 1\} \end{cases} \tag{5}$$

The first dimension of the embedding applies a linear projection to time, followed by $d_\tau - 1$ periodic functions where the period and initial phase are learnable.

3.2 CROSS-DIMENSION TRANSFORMER FOR IRREGULAR TIME-SERIES

To deal with the temporal irregularity of observations with missing values, we equip the cross-dimension transformer block (Kong et al., 2020), which has been also adapted by the CSDI paper (Tashiro et al., 2021), with our learnable time embedding. Each cross-dimension transformer block has two attention mechanisms, one across all the actual measurements over the feature dimension for each observation and another one across all the observations over the time dimension for each feature. Multiple cross-dimension transformer blocks with residual connections are stacked together as the major component in our model's implementation. To apply this cross-dimension transformer block, we first represent feature type, time of the event, and masking information for each value in $\boldsymbol{X}_s$ with the following vector,

$$\boldsymbol{R}[i,k] = concat(F_k, E(\tau_i), \mathbb{1}\{(i,k) \in co\}). \tag{6}$$

In this representation, F_k is a learnable embedding of size d_f for the kth feature in measurements, E is the learnable time embedding defined in Sec. 3.1 and $\mathbb{1}$ is an indicator function, resulting a time-feature-mask representation $\boldsymbol{R}$ of shape $L \times K \times (d_f + d_\tau + 1)$. Let h be the hidden dimension of a standard transformer block, first with a linear map and $H^{(r-1)} \in \mathbb{R}^{L \times K \times h}$ be the output of the $r-1$th block. We update $H^{(r-1)}$ to obtain $H^{(r)}$ through the following steps:

$$H'^{(r)} = H^{(r-1)} + \boldsymbol{R} W_E^{(r)} + \mathcal{D}[s], \tag{7}$$

$$H^{*(r)}[:,k] = \phi^{(r)}(H'^{(r)}[:,k]), \tag{8}$$

$$H^{(r)}[i] = \psi^{(r)}(H^{*(r)}[i]), \tag{9}$$

where $W_E^{(r)} \in \mathbb{R}^{(d_f + d_\tau + 1) \times h}$ is a learnable projection matrix, and $\mathcal{D}[s] \in \mathbb{R}^h$ is a learnable embedding of the diffusion step s passing noise scale information to the model. $\phi^{(r)}$ and $\psi^{(r)}$ in Eq. 8 and Eq. 9 are regular transformer encoders (Vaswani et al., 2017) that take the columns of $H'^{(r)}$

and rows of $H^{*(r)}[i]$ respectively. $H^{(0)}$ is defined as a linear map of the $\boldsymbol{X}_s$, mapping each single value to a vector of size h. There are Swish activation functions (Ramachandran et al., 2017) between the transformer blocks.

3.3 Representation for Downstream Recognition Tasks

To derive representations for downstream tasks, we must first extract intermediate states from the pretrained backbone model. These states are then fed into task-specific heads for recognition tasks. However, adapting a conditional diffusion model as a representation extractor presents a challenge due to its inherent design: it assumes noisy input and produces stochastic output, whereas a deterministic output is preferred for representation. To address this discrepancy, we keep all the observed values in the context set co, use diffusion step embedding at the minimum noise level $\mathcal{D}[0]$, and refrain from injecting any noise into the data. In our experiments, adding any level of noise or a higher degree of diffusion led to diminished performance, contrasting with the outcomes observed in the DiffMAE model. We hypothesize that this discrepancy is a result of the substantial information redundancy in images, which is not the case in the irregular time-series datasets we explored.

The output of each cross-dimensional transformer block is a hidden state of shape $L \times K \times h$. The hidden states across all transformer blocks are concatenated along the last dimension for maximum possible representation power. The concatenated tensor is summarized by multi-layer perceptrons and mean pooling to obtain representation vectors which are then fed to a linear classification layer. Mean pooling along the time dimension will be omitted if we need to make predictions for each individual time step.

3.4 Complexity Analysis

Diffusion models are known for their high computational costs. However, the generation task from pure noise typically requires most of these sequential steps, whereas our approach for the downstream task bypasses this requirement by employing the denoising function just once. The complexity of each layer for a sample during training is $O(KLh^2(K+L))$. This complexity arises from two Transformer layers: one across the time steps and the other across the features, each repeated across every feature and time step accordingly. While the generation process demands S sequential repetitions of this layer, our approach for using the pretrained model for a discriminative task eliminates the need for these S steps, reducing the number of sequential operations to $O(1)$.

4 Experiments

4.1 Datasets and Tasks

We use the PhysioNet (Silva et al., 2012) and Human Activity (Kaluža et al., 2010) datasets in our experiments. PhysioNet contains medical measurements conducted on the patients in their first two days in the ICU and an in-hospital mortality label for each patient. The measurements are asynchronous and taken with different frequencies for different patients. Sequences in the dataset have many missing values as measurements are asynchronous and are recorded with different frequencies for the patients. The final task is a binary classification, predicting the in-hospital mortality of each individual. Classes are highly unbalanced in this dataset.

The Human Activity dataset consists of measurements from four sensors attached to five subjects' bodies and each sensor has spatial measurements (x, y, z) measured irregularly. As followed by (Rubanova et al., 2019), we break long sequences into sequences of length fifty, using a sliding window across the sequence with an overlap of size 25 between two consecutive collected subsamples. The task is to predict the type of activity of the person at each time of the sequence from six possible activities.

4.2 Results

As our back-bone model is trained on a conditional generation task, we compare it against other conditional generative models for time series in Table 1 with three fixed missing ratios. We report CRPS (Matheson & Winkler, 1976; Tashiro et al., 2021) as an evaluation metric for probabilistic

Table 1: CRPS measurements compared for the PhysioNet (Kaluža et al., 2010) dataset with different missing ratios (lower is better).

	10% missing	50% missing	90% missing
Latent ODE (Rubanova et al., 2019)	0.700	0.676	0.761
mTANs (Shukla & Marlin, 2021)	0.526	0.567	0.689
CSDI w pos encoding (Tashiro et al., 2021)	0.380	0.418	0.556
Ours	**0.256**	**0.291**	**0.542**

Table 2: The table contains imputation and classification results on PhysioNet and Activity.

Model	PhysioNet (Silva et al., 2012)			Activity (Kaluža et al., 2010)		
	AUC	RMSE	CRPS	Accuracy	RMSE	CRPS
mTAN (Shukla & Marlin, 2021)	0.837	6.89	0.567	91.0	20.46	0.164
PrimeNet (Chowdhury et al., 2023)	0.842	**4.78**	N/A	89.9	14.30	N/A
Ours	**0.855**	5.32	**0.292**	**92.4**	**14.19**	**0.142**

models. The results of this experiment indicate that replacing learnable time embeddings with positional encodings can significantly improve the performance of the model for irregular time series. The improvement is more significant when more time steps are available. For this table, we use the baseline numbers from Tashiro et al. (2021).

We further compare our models with baseline models on classification and imputation tasks in Table 2. We use a fifty percent missing ratio for the PhysioNet dataset and ten percent for the Human Activity dataset for the imputation task, following the experimental setup in Chowdhury et al. (2023). Due to class label imbalance, the metric for the classification task on the PhysioNet dataset is the area under ROC curve (AUC). We use the classification accuracy metric for the Human Activity dataset. We report RMSE of point estimate results for all models and CRPS for probabilistic models including our models and mTAN. We calculate the RMSE by one hundred times sampling from the distribution and getting the median. As a probabilistic approach, our method focuses on learning a distribution, which may not match the effectiveness of top point estimation techniques for this specific purpose.

We follow the same data split as our baselines, using dataset splits from Tashiro et al. (2021) for Table 1 and splits from Shukla & Marlin (2021) for Table 2. In all of the experiments we use twenty percent of the samples as the test set, and use twenty percent of the remaining as a validation set and all remaining as the train set.

5 Conclusion and Discussions

In this work, we expanded the capabilities of conditional diffusion models for irregular time series by integrating learnable time encodings instead of traditional positional encodings. Inspired by the masking techniques for sequential data from extensive existing works, we proposed using conditional diffusion models as a generalized alternative to masking approaches for time series analysis. Using a spectrum of noise levels enriches the learning process, enabling larger models to train on small datasets without overfitting. However, this diversity in the training task means the training process may take longer time. Our experiments show that conditional diffusion models are effective as self-supervised training, capturing rich features that benefit downstream tasks in irregular time series. Although our model realization uses the same architecture as CSDI Tashiro et al. (2021), integrating a wider range of architectures with this pre-training strategy presents a promising direction for future research aimed at enhancing model efficiency, time and space complexity, and scalability. Furthermore, applying insights and techniques from training diffusion models, such as using advanced ODE solvers, could further advance this area of work.

REFERENCES

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

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *arXiv preprint arXiv:2302.00861*, 2023.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. *arXiv preprint arXiv:2312.02139*, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Boštjan Kaluža, Violeta Mirchevska, Erik Dovgan, Mitja Luštrek, and Matjaž Gams. An agent-based approach to care in independent living. In *Ambient Intelligence: First International Joint Conference, AmI 2010, Malaga, Spain, November 10-12, 2010. Proceedings 1*, pp. 177–186. Springer, 2010.

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

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Zhe Li, Zhongwen Rao, Lujia Pan, Pengyun Wang, and Zenglin Xu. Ti-mae: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

David Lüdke, Marin Biloš, Oleksandr Shchur, Marten Lienen, and Stephan Günnemann. Add and thin: Diffusion for temporal point processes. *arXiv preprint arXiv:2311.01139*, 2023.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

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

Satya Narayan Shukla and Benjamin M Marlin. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318*, 2021.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

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

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoder. In *ICCV*, 2023.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023.

A DATASETS

A.1 PHYSIONET

The PhysioNet dataset features health-related measurements from ICU patients within their initial 48 hours of admission. Introduced during a challenge, it includes 12,000 samples, equally distributed across train, test, and validation subsets, each containing 4,000 samples. Initially, labels were only released for the training data and samples from the validation set. Currently, labels for all subsets are accessible; however, our experiments adhere to the experimental setup outlined in Chowdhury et al. (2023); Tashiro et al. (2021). Specifically, Table 1 uses the configuration from Tashiro et al. (2021), using only training samples from the original dataset. In contrast, Table 2 follows the approach of Chowdhury et al. (2023), merging the validation and train sets for the imputation task but only using the train set for classification. The dataset has 36 time-related and 5 static variables—age, gender, ICU type, height, and weight. For classification, we combine static features with time-related data, treating the static variables as if measured only at time zero. Alternatively, these static features could be excluded from the time-series analysis and incorporated with a linear mapping into the final classification layer. In all tasks, we split the dataset with a ratio of with a split ratio of 64/16/20 for the train, validation, and test subsets, respectively.

A.2 HUMAN ACTIVITY

The Human Activity dataset was collected from five individuals performing various types of activities. Although the original dataset includes 11 activity types, it is customary to consolidate these into six categories due to the high level of similarity among some classes (Rubanova et al., 2019). Each participant was equipped with four sensors, each measuring an (x, y, z) spatial location. The measurements are irregular, with each measurement potentially capturing data from one or more sensors. Consequently, at each time-step, we obtain a feature $x_i \in \mathbb{R}^{12}$, representing the aggregated sensor data. In most of the time steps, just measurements from a single sensor are available, making the feature matrix $\boldsymbol{X}$ mostly sparse. Following the preprocessing method of our baseline, we generated sequences of length 50 from the longer activity sequences, with consecutive sequences sharing a 25 time-step overlap. Ultimately, the dataset includes 6,554 samples, which were randomly split into train, validation, and test sets using a 64/16/20 ratio.

B EXPERIMENTS DETAILS

C PRETRAINING PROCESS

During each epoch of pretraining, a subset of observed values is selected for noise addition. This selection involves randomly sampling a masking ratio $0 < p < 1$ from a uniform distribution for each training iteration. Then, a rounded pn number of values from the n observed values are chosen, and a diffusion step s is uniformly selected from 0 to S, with S representing the total number of diffusion steps. These selected values are then subjected to noise, as defined by equation 2, and subsequently denoised, conditioned on the remaining clean observed values. Unlike the gradual denoising in the generation/imputation phase, the training process targets direct estimation of the original data from the noised values in each iteration, reminding the conventional masking techniques for sequential data.

C.1 HYPERPARAMETERS

For the diffusion process, we follow the same setup as CSDI, using 50 steps for the diffusion process, values of $\beta_1, \beta_2, \cdots \beta_{50}$ change quadratically from 0.0001 to 0.5. We emphasize that this choice is significantly smaller than 1000 steps used in the original DDPM method used for images (Ho et al., 2020). However, it shows promising results in our case. Increasing the number of steps will make the training process and generative model much slower.

For hyperparameter optimization, we experimented with both Adam (Kingma & Ba, 2014) and AdamW (Loshchilov & Hutter, 2017) optimizers, employing step scheduling and the cosine learning rate scheduler (Loshchilov & Hutter, 2016). During pretraining, we conducted a grid search over the

following hyperparameters: hidden dimension (64, 128, 256), number of residual blocks (4, 6, 8), and learning rate (0.01, 0.001, 0.0001). The PhysioNet dataset was trained for 350 epochs, while the Human Activity dataset underwent 1000 epochs of training, reflecting our model's need for a larger capacity to handle more generalized tasks than masked autoencoders. Despite larger models yielding better performance, computational constraints limited our model size. For classification model training, a similar approach was taken for learning rate optimization, with a linear search determining that a hidden dimension of 64 was optimal for the PhysioNet dataset, whereas for the Human Activity dataset, we used a hidden dimension of size 400.

C.2 Classification Head

The output of each residual block contains a representation vector for each value in each time-step, with a shape of $B \times K \times h$. Unlike autoencoders where encodings aim to create a compressed view of the data, our encoding results in a vector even larger than the original data.

To summarize these mappings for the downstream task, our classification head design involves initial summarization across features and then across time. This process includes a linear transformation, an activation function, and mean pooling. A final linear function maps the resulting embeddings to classification logits. For the Human Activity dataset, classification is performed across each time-step, eliminating the need for the second pooling across time. Instead, a linear mapping is retained to maintain the same complexity in the downstream head as before, measured in the number of layers.

C.2.1 Training Classification Head

In the fine-tuning for the classification task, we exclusively update the weights for the classification head, keeping the weights of the frozen backbone model unchanged. This approach prevents overfitting, as the pretraining model allows training a much larger model on a small dataset. The backbone model's capacity is substantial, and even a single epoch of fine-tuning with a small learning rate can lead to overfitting. Hence, we maintain the frozen state of the backbone model during fine-tuning.