DIFFODE: Neural ODE with Differentiable Hidden State for Irregular Time Series Analysis

Yudong Zhang[1,2], Xu Wang[1,2*], Xuan Yu[1], Zhengyang Zhou[1,2], Xing Xu[1], Lei Bai[3], Yang Wang[1,2*]
[1]University of Science and Technology of China (USTC), Hefei, China
[2]Suzhou Institute for Advanced Research, USTC, Suzhou, China
[3]Shanghai AI Laboratory, Shanghai, China
{zyd2020@mail., wx309@, yx2024@mail., zzy0929@, angyan@}ustc.edu.cn,
star.xussi@gmail.com, baisanshi@gmail.com

2025 IEEE 41st International Conference on Data Engineering (ICDE) | 979-8-3315-3603-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICDE65448.2025.00160

***Abstract*—Irregular time series analysis is increasingly essential in data management due to the proliferation of complex data irregularly sampled by real-world systems. Traditional time series models, including RNN-based models and transformer variants, face significant challenges in generalizing to continuous-time paradigms, which are essential for capturing the ongoing dynamics of irregular time series. Neural Ordinary Differential Equations (NODEs) assume a continuous latent dynamic and provide an elegant framework for irregular time series analysis, yet they suffer from limitations like fragmented latent processes and the inability to fully exploit interdependencies among observations. To address these challenges, we propose a novel Differentiable hidden state enhanced neural ODE framework, termed DIFFODE, designed to effectively model irregular time series. Concretely, we introduce an attention-based differential hidden state that maps irregular observations into a continuous hidden state space, enabling the extraction of latent dynamics while preserving temporal continuity. Leveraging the theory of generalized inverses, DIFFODE innovatively derives ODEs to describe hidden state dynamics. Furthermore, we incorporate the Hoyer metric into our framework to enhance its capacity to capture subtle yet critical temporal shifts, significantly improving the accuracy of time series modeling. Extensive experiments on both synthetic and real-world datasets demonstrate the effectiveness of DIFFODE across three key tasks, including irregular time series classification, interpolation, and extrapolation.**

***Index Terms*—Neural ODEs, irregular time series analysis, differentiable hidden state.**

I. Introduction

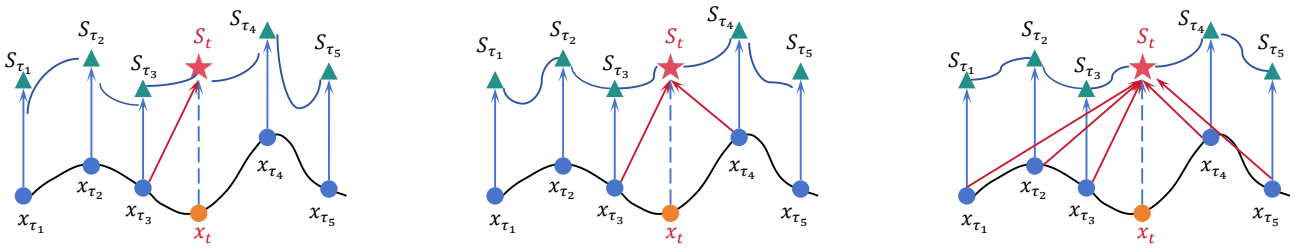
Irregular time series analysis has become increasingly significant in data management due to the proliferation of complex temporal data generated by modern real-world systems [1]–[5]. From IoT devices and industrial sensors to financial transactions and healthcare records [6]–[10], time series data in many scenarios frequently exhibit irregularities caused by event-driven processes, sensor malfunctions, or varying data collection frequencies, leading to difficulty in unifying such data into consistent time intervals. However, existing data management methods struggle to store, process, and analyze such incomplete, sparse, or non-uniform time-series data, which makes the potential of data under these non-ideal conditions not fully unlocked [11]–[15]. As irregular time series datasets become growing prevalent across diverse domains, effectively understanding and handling irregular time series is essential to improving data-driven decision-making and enabling more robust data management.

With the rapid advancement of deep learning, its application in time series analysis has garnered increasing attention and achieved remarkable performance across various tasks [16]–[22]. Recurrent neural network (RNN) and its variants [23]–[25] are classical sequential models that often require explicit preprocessing (*e.g.*, interpolation) to handle irregular timestamps, which can distort temporal dynamics, while their recurrent structure struggles with long-term dependencies. Regarding attention-based models [26]–[30], despite their strength in capturing long-range dependencies, they are limited by their reliance on fixed-length representations, high data demands, and computational inefficiency for irregular sequences. Additionally, State Space Model (SSM)-based approaches [31], [32] have also been studied and offer a probabilistic framework for temporal modeling, but they often rely on strong assumptions and can become computationally intractable for complex datasets. In short, these approaches share a common limitation: their inability to natively model continuous temporal evolution and irregular sampling patterns. This motivates the adoption of Neural Ordinary Differential Equations (NODEs), which can model the continuous dynamics in irregular timestamps through adaptive solvers and has become a mainstream approach for irregular time series analysis [33], [34].

Although NODE-based methods [35]–[42] are theoretically effective for modeling continuous dynamics, they exhibit critical limitations when applied to irregular time series. These methods typically integrate from an initial value to derive all subsequent values, without considering observed data points later than the initial point. They integrate the latent state at each time point with observations, *i.e.*, having different initial values at different time intervals. While such a mechanism can achieve a certain level of accuracy, it considers only one observation at a time and neglects the interdependencies among observations. Consequently, this mechanism results in a *fragmented latent process* that may fail to accurately represent the true underlying dynamics, as illustrated in Fig. 1 (a). To tackle the issue of the fragmented latent state of NODEs, Neural Controlled Differential Equations (NCDE) approach [43]–[45] offers an

* Corresponding authors: Yang Wang, Xu Wang.

2375-026X/25/$31.00 ©2025 IEEE
DOI 10.1109/ICDE65448.2025.00160



(a) Neural ODE (b) Neural CDE (c) Ours

Fig. 1: An illustration of Neural ODE, Neural CDE, and our method. For a given time point, *NODE* integrates from the last observation, resulting in a fragmented latent process. *NCDE* employs an interpolation algorithm to calculate a continuous path, but fails to fully exploit the contextual information in the data. *Our method* introduces an attention-based differential hidden state, which adeptly captures temporal dynamics while ensuring the seamless continuity of the latent process.

alternative by interpolating the observed values to estimate a continuous latent process. For instance, the use of natural cubic spline interpolation in [43] allows the model to construct a latent trajectory that guides the integration path, thereby incorporating subsequent observations into the modeling process. However, despite their simplicity, these methods *fail to fully exploit the contextual information inherent in the data.* As depicted in Fig. 1 (b), such methods typically rely only on the two nearest observations at any given time point, which limits their ability to capture long-range dependencies. Furthermore, interpolation algorithms, while useful for estimating intermediate states, are limited in their ability to capture the temporal correlations intrinsic to the time series, thereby constraining the model's capacity to represent the true dynamics comprehensively.

To address the limitations of existing approaches, this paper introduces a novel **Diff**erentiable hidden state enhanced neural **ODE** framework, termed **DIFFODE**, a data-driven solution designed to effectively capture complex temporal dynamics while ensuring the seamless continuity of the latent process. Specifically, to fully leverage the potential of continuous dynamics in irregularly sampled data, we design an *attention-based differential hidden state* as depicted in Fig. 1 (c), which treats irregular observations as a projection matrix that maps the time series into a hidden state space. The linearity of this projection ensures that the hidden states preserve the continuity inherent in the original time series. Building on this foundation, our framework utilizes the theory of generalized inverses to reverse-engineer the attention mechanism, thereby deriving ODEs that describe the dynamics of the hidden states. Furthermore, to enhance the precision in modeling temporal relationships, we incorporate the *Hoyer metric* [46], an advanced tool of sparsity metric. By strategically maximizing the Hoyer metric, our framework sharpens its ability to discern subtle yet significant temporal shifts, thereby improving the accuracy and reliability of predictions. Finally, we conduct extensive experiments on both synthetic and real-world datasets to evaluate the proposed DIFFODE across irregular time series classification, interpolation, and extrapolation tasks.

The main **contributions** of our work are summarized in the following four areas:

- *Novel insight and framework:* We identify the critical challenges of continuous dynamic representation in irregular time series analysis, and innovatively propose a new data-driven neural ODE framework (DIFFODE) to unleash the great potential inherent in time series data, thereby driving the applications of data management to more diverse and complex data scenarios.
- *In-depth dynamics extraction:* We devise an attention-based differential hidden state to effectively capture the continuous dynamics of the latent process, and leverage the theory of generalized inverses to derive ODEs that describe the hidden state dynamics of irregular time series.
- *Precise temporal modeling:* We delicately incorporate the Hoyer metric into our framework, which enhances the ability of DIFFODE to discern subtle yet significant temporal shifts, significantly improving the accuracy and reliability of the modeling process.
- *Compelling empirical validation:* Extensive experiments conducted on both synthetic and real-world datasets validate the effectiveness of DIFFODE across the mainstream irregular time series tasks, including classification, interpolation, and extrapolation.

II. Related Work

We briefly review the related literature below, including time series analysis with deep learning and neural ODEs for irregular time series.

A. Time Series Analysis with Deep Learning

Time series analysis is a foundational problem in the field of data engineering, with applications spanning finance, healthcare, climate science, and urban systems [47]–[53]. Deep learning has shown significant potential in modeling complex temporal dependencies in time series data. Recurrent Neural Networks (RNNs) [26], [29], [30], [54], [55] were among the first neural architectures applied to sequential data, followed by Long Short-Term Memory networks (LSTMs) [56] and Gated Recurrent Units (GRUs) [57], which address the vanishing gradient problem and enable the modeling of long-range dependencies. However, RNN-based models often suffer from high computational overhead and difficulties in parallelization, leading

researchers to explore alternative architectures. Convolutional Neural Networks (CNNs), while traditionally used for image data, have been adapted for time series analysis. Temporal Convolutional Networks (TCNs) [58], for example, extend CNNs by incorporating causal convolutions and dilations, enabling the modeling of long-term dependencies while preserving temporal causality. The Transformer model [59] has brought transformative changes to time series analysis. Adaptations such as Informer [26] and iTransformer leverage attention mechanism to capture interactions and dependencies among multiple variables in multivariate time series. Hybrid models combining multiple architectures have emerged to address these challenges [60], [61]. For example, CNN-Transformer hybrids [62] aims to integrate local feature extraction capabilities of CNNs with the global dependency modeling of Transformers. Neural Ordinary Differential Equations (Neural ODEs) [33] are another recent advancement, offering a continuous-time framework for modeling irregularly sampled time series, particularly in domains like healthcare [35].

Thus, studies that utilize Neural ODEs to model continuous-time processes for irregular time series [33] are much more closely aligned with our work, and we provide a detailed description of this paradigm below.

B. Neural ODEs for Irregular Time Series

Neural Ordinary Differential Equations (NODEs) have gained considerable attention in the analysis of irregular time series due to their remarkable capability to capture temporal dynamics. Existing NODE-based approaches can be broadly categorized into two types. The first category involves methods that rely on interpolating or approximating latent dynamics to model time series. Neural Controlled Differential Equations (NCDE) [63] employ rough path theory to model long-term sequence dependencies more effectively. ContiFormer [44] extends the Transformer framework to continuous time by using interpolation to construct queries for its attention mechanism. Neural LAD [45] further advances this line of research by modeling periodicity, trends, and local information, with local dynamics derived from the differential of interpolated sequences. However, methods in this category often generate the entire time series from a single initial value and fail to fully exploit the rich contextual information present in the data. The second category focuses on incorporating discrete updates into NODEs to account for irregular observations. ODE-RNN [35] and ODE-LSTM [36] use gating mechanisms to update latent states with new information. CADN [37] builds upon the ODE-RNN framework by integrating an attention mechanism for enhanced modeling. GRU-ODE-Bayes [38] and Neural Jump ODE [8] adopt Bayesian estimation techniques in their update steps. GNODE [40] and TGNN4I [41] extend NODEs to graph structures, employing graph neural networks (GNNs) combined with GRUs to model latent state changes. ANDE [42] introduces the HiPPO matrix into the integral step to enhance the representation of historical information, updating directly via assignment in the update step.

Nevertheless, existing methods still have limited ability to fully capture the underlying continuous dynamics of irregular time series data, which is exactly a critical challenge to be addressed in this work.

TABLE I: Key symbols and corresponding descriptions.

Symbol	Description
X_{ob}	Observed irregular time series, including time points and values.
T_{ob}	Set of observed time points: $T_{ob} = \{t_1, t_2, \ldots, t_n\}$.
x_t	Value of the time series at time t.
z_t	Latent representation of x_t, generated by a neural network.
y_t	Output of the time series at time t.
Z	Latent representations for all observations: $Z = [z_{\tau_1}, z_{\tau_2}, \ldots, z_{\tau_n}]^\top$.
S_t	Differentiable hidden state (DHS) at time t, representing continuous dynamics.
p_t	Attention scores for z_t relative to all observations Z.
a_t	Unnormalized attention scores between z_t and Z: $a_t = \frac{z_t Z^\top}{\sqrt{d}}$.
P_{diag}	Diagonal matrix formed from p_t, used in computing the dynamics of S_t.
$\phi(\cdot)$	Neural network modeling the time derivative $\frac{\mathrm{d}z_t}{\mathrm{d}t}$.
$F_s(\cdot)$	Differential equation governing the dynamics of S_t.
$(Z^\top)^\dagger$	Moore-Penrose inverse of $Z^\top$, used to compute p_t backward.
$\text{Hoyer}(\cdot)$	Sparsity metric used to measure and optimize the sparsity.

III. Methodology

In this section, we begin by formulating the problem of time series modeling using ordinary differential equations. Subsequently, we introduce the concept of a Differentiable Hidden State (DHS) as the foundation of our approach, and provide a detailed explanation of our proposed framework. Finally, we elaborate on the key derivations and optimization processes that underpin our method. For clarity, Table I summarizes the frequently used symbols and their corresponding descriptions throughout this paper.

A. Modeling Time Series with ODEs

We denote the irregular time series of interest as $X_{ob} = \{(x_t, t) | x_t \in \widetilde{X}, t \in T_{ob}\}$, where observations $\widetilde{X} = \{x_{\tau_1}, x_{\tau_2}, \cdots, x_{\tau_n}\}$ are sampled irregularly at time points $T_{ob} = \{t_1, t_2, \cdots, t_n\}$, and n is the number of observations. These observations are assumed to be sampled from an underlying continuous time series $X_{co} = \{x_t | x_t, t \in \mathbb{R}\}$.

To model the continuous dynamics of the hidden states corresponding to this irregular time series, we utilize an Ordinary Differential Equation (ODE) framework. Specifically, the dynamics are expressed as:

$$\frac{\mathrm{d}S_t}{\mathrm{d}t} = F_s(S_t, X_{ob}, t), \tag{1}$$

where S_t denotes the hidden state of the time series at time t and $F_s(\cdot)$ governs the evolution of the hidden state based on its current value, observed data, and time.

For any arbitrary time t, the hidden state S_t can be computed by integrating 1 from an initial time point τ_1 with an initial hidden state S_{τ_1}:

$$S_t = S_{\tau_1} + \int_{\tau_1}^{t} F_s(S_\tau, X_{ob}, \tau) d\tau. \tag{2}$$

Finally, a readout function f_{out} is applied to S_t to generate the corresponding output of the time series at time t:

$$y_t = f_{out}(S_t). \tag{3}$$

B. Differentiable Hidden State Based on Discrete Observations

In this paper, we propose a **D**ifferentiable **H**idden **S**tate (**DHS**) S_t as the continuous dynamics of time series in Eq. 1. The proposed DHS is generated from the latent representations of time series, which encodes values of time series and their corresponding time points. Specifically, given any time point t and corresponding data x_t, the latent representation z_t is obtained by a neural network,

$$\psi : (x_t, t, E(x_t)) \to z_t, \tag{4}$$

where $E(x_t)$ refers to the external features corresponding to x_t. In practice, we find introducing historical observations of x_t when obtaining z_t leads to better performance, *i.e.*, we have $E(x_t)$ as $\{x_i | i < t\}$. Therefore, latent representations on all observation time points can be denoted as $Z = [z_{\tau_1}, z_{\tau_2}, \cdots, z_{\tau_n}]^\top \in \mathbb{R}^{n \times d}$.

An attention mechanism is applied to generate the differentiable hidden state. Let z_t be Query, and Z be Key and Value. Then we define DHS as

$$a_t = \frac{z_t Z^\top}{\sqrt{d}},\ p_t = \mathrm{Softmax}(a_t),\ S_t = p_t Z, \tag{5}$$

where $a_t, p_t \in \mathbb{R}^n, S_t \in \mathbb{R}^d$ and we always have $n > d$. Here, a_t is the attention score and indicates the correlations between data at time t and other time, and p_t is the normalization of it. DHS is defined on all observations according to correlations with them.

The above definition of DHS suggests that one can obtain a continuous state space of a time series, as illustrated in Figure 2, where the hidden state S_t at any time t is correlated to the latent representation z_t of time series at t and the latent representations Z of all irregularly sampled observations X_{ob}. Based on the definition of DHS, the derivative of DHS can be calculated, and the differential equation describing the dynamics of DHS can be obtained.

C. Execution Process of DHS

In this section, we aim at achieving the differential equation of DHS as defined in Eq. 1, while giving the detailed form of F_s. According to Eq. 5, the derivative of DHS S_t with respect to time t can be calculated using chain rule as,

$$\frac{\mathrm{d}S_t}{\mathrm{d}t} = \frac{\mathrm{d}z_t}{\mathrm{d}t} \frac{Z^\top (P_{diag} - p_t^\top p_t) Z}{\sqrt{d}}, \tag{6}$$

where $P_{diag} = \mathrm{Diag}(p_t)$, and $p_t = [p_{t,1}, p_{t,2}, \cdots, p_{t,n}]$ corresponds to normalized attention score of z_t to all observations Z as in Eq. 5. The detailed calculation flow of the above process is as follows: for simplicity, let $z_{\tau_i} = z_i$, so that $Z = (z_1^\top, z_2^\top, \cdots, z_n^\top)^\top$. Noting that $\forall i$, z_i is independent of t. The derivative of softmax is

$$\frac{\partial p_j}{\partial a_i} = \begin{cases} p_j(1 - p_j), & i = j \\ -p_i p_j, & i \neq j \end{cases}. \tag{7}$$

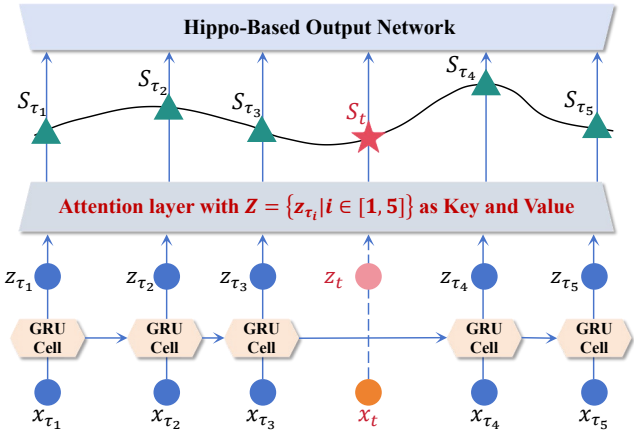


Fig. 2: Solution overview. Irregularly sampled observations are fed into a neural network to generate Z, which serves as the key and value of the attention layer to generate a differentiable hidden state. A HiPPO-based output network is employed to generate the output of the whole framework.

Next,

$$\frac{dS_t}{dt} = \sum_{j=1}^{n} \left(\frac{dp_j}{dt} z_j + p_j \frac{dz_j}{dt}\right) = \sum_{j=1}^{n} \frac{dp_j}{dt} z_j, \tag{8}$$

where

$$\begin{aligned} \frac{dp_j}{dt} &= \sum_{i=1}^{n} \frac{\partial p_j}{\partial a_i} \frac{da_i}{dt} = p_j(1 - p_j)\frac{da_j}{dt} - \sum_{i \neq j} p_i p_j \frac{da_i}{dt} \\ &= p_j \frac{da_j}{dt} - \sum_{i=1}^{n} p_i p_j \frac{da_i}{dt} \\ &= p_j \frac{dz_t}{dt}\frac{z_j^\top}{\sqrt{d}} - \sum_{i=1}^{n} p_i p_j \frac{dz_t}{dt}\frac{z_i^\top}{\sqrt{d}}. \end{aligned} \tag{9}$$

Then,

$$\begin{aligned} \frac{dS_t}{dt} &= \frac{dz_t}{dt}\left(\sum_{i=1}^{n} p_i \frac{z_i^\top z_i}{\sqrt{d}} - \sum_{i=1}^{n}\sum_{j=1}^{n} p_i p_j \frac{z_i^\top z_j}{\sqrt{d}}\right) \\ &= \frac{dz_t}{dt}\frac{1}{\sqrt{d}}(Z^\top (P_{diag} - p_t^\top p_t) Z), \end{aligned} \tag{10}$$

where $P_{diag} = \begin{pmatrix} p_1 & & \\ & \ddots & \\ & & p_n \end{pmatrix}$.

Since the first term $\frac{\mathrm{d}z_t}{\mathrm{d}t}$ in Eq. 6 is intractable, we follow the NODE framework [33] and approximate it with a neural network ϕ:

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = \phi(z_t, t). \tag{11}$$

Substituting this into the derivative of S_t, we obtain:

$$\frac{\mathrm{d}S_t}{\mathrm{d}t} = \phi(z_t, t) \frac{Z^\top (P_{diag} - p_t^\top p_t) Z}{\sqrt{d}}. \tag{12}$$

To achieve a differential equation of S_t as in Eq. 1, given observations X_{ob}, the derivative of S_t should be only dependent

on S_t and t. However, in Eq. 12, while Z is transformation of X_{ob}, $\frac{\mathrm{d}S_t}{\mathrm{d}t}$ are dependent on p_t and z_t. In the following, we further transform p_t and z_t into S_t by innovatively computing the attention mechanism backward.

Note that in Eq. 5, the dimension of p_t is higher than that of S_t, thus the information is compressed in this step. If we consider equation Eq. 5 as a linear system and solve it directly, we will get infinite solutions. To attain a proper S_t, we introduce the theory of generalized inverse. Generalized inverse allows for a unified approach to obtaining solutions for linear systems, no matter how many solutions they may have. We give the detailed definition of generalized inverse as follows:

Definition 1. *(Generalized Inverse) [64], [65]. The purpose of constructing a generalized inverse matrix is to obtain a matrix that can serve as an inverse in some sense for a wider class of matrices than invertible matrices. Suppose $A \in C^{m\times n}$ is any complex matrix, if there exists a complex matrix $G \in C^{n\times m}$ such that at least one of the following conditions holds: i) $AGA = A$, ii) $GAG = G$, iii) $(AG)^H = AG$, iv) $(GA)^H = GA$. Then G is called a generalized inverse matrix, and the four equations above are called Moore-Penrose (M-P) equations. Furthermore, G is called the Moore-Penrose inverse of A if G satisfies all of the four M-P equations, denoted as $G \in A\{1,2,3,4\}$. In general, if G satisfies the i_1-th, i_2-th, $\cdots$, i_k-th ($1 \le k \le 4$) one of the four M-P equations, then G is a weak inverse of A, denoted as $G \in A\{i_1, i_2, \cdots, i_k\}$.*

Usually there exists different notations for the commonly used generalized inverse: $A\{1\}$ is called the minus sign inverse, denoted as A^-; $A\{1,2\}$ is called the reflecsive minus sign inverse, denoted as A_r^-; $A\{1,3\}$ is called the least square generalized inverse, denoted as A_l^-; $A\{1,4\}$ is called the least norm generalized inverse, denoted as A_m^-; $A\{1,2,3,4\}$ is called the Moore-Penrose inverse, denoted as $A^\dagger$.

In our case, the solution for p_t could be expressed as

$$p_t^\top = (Z^\top)^\dagger S_t^\top + (I_n - (Z^\top)^\dagger Z^\top)h, \tag{13}$$

where h is a random vector of dimension n, and $(Z^\top)^\dagger$ is the Moore-Penrose inverse [64] of $Z^\top$. In most cases, we have $n \gg d$ holds, so we can assume that $Z^\top$ has full row rank and thus have $(Z^\top)^\dagger = Z(Z^\top Z)^{-1}$.

According to the theory of generalized inverse, we could readily obtain the minimum-norm solution $p_t^\top = (Z^\top)^\dagger S_t^\top$. However, a more appropriate solution could be attained by considering the properties of p_t.

In the attention mechanism, p_t is always sparse so as to concentrate on certain important time points. We introduce Hoyer [46] to measure the sparsity of p_t.

Definition 2. ***(Hoyer Sparsity Metric) [46].*** *Given a vector $x \in \mathbb{R}^N$, Hoyer could be defined as*

$$\mathrm{Hoyer}(x) = \frac{1}{\sqrt{N}-1}(\sqrt{N} - \frac{\sum_{i=1}^N x_i}{\sqrt{\sum_{i=1}^N x_i^2}}). \tag{14}$$

As a measure of sparsity, Hoyer has several excellent properties [46]:

(a) $\forall \alpha, x_i, x_j$ such that $x_i > x_j, 0 < \alpha < \frac{x_i - x_j}{2}$, we have $\mathrm{Hoyer}([x_1, \cdots, x_i - \alpha, \cdots, x_j + \alpha, \cdots]) < \mathrm{Hoyer}(x)$.
(b) $\forall \alpha \in \mathbb{R}, \alpha > 0$, we have $\mathrm{Hoyer}(\alpha x) = \mathrm{Hoyer}(x)$.
(c) $\forall i, \exists \beta > 0$, such that $\forall \alpha > 0$, we have $\mathrm{Hoyer}([x_1, \cdots, x_i + \beta + \alpha, \cdots]) > \mathrm{Hoyer}([x_1, \cdots, x_i + \beta, \cdots])$.
(d) $\mathrm{Hoyer}(x||0) > \mathrm{Hoyer}(x)$, where $||$ denotes concatenation.

Criterion (a) implies that if the sum of the vector remains constant, then the more uniformly distributed, the less sparse the vector will become. Criterion (b) suggests that sparsity is a relative property. Multiplying all elements by the same factor does not alter the sparsity. Criterion (c) finds a main element. When the main element is large enough, it is able to determine the sparsity of the vector. Criterion (d) naturally follows from the definition of sparsity. According to these properties, we can conclude that *the larger the value of* $\mathrm{Hoyer}(\cdot)$*, the sparser the vector.*

From Eq. 12, a proper vector h is required to get a sparse p_t. We construct an optimization problem based on Hoyer. Note that p_t is the result of softmax normalization, so the elements are all positive and the sum of them is 1. Let $J_{1,n}$ and $J_{n,1}$ denote all-one matrices of dimension $1 \times n$ and $n \times 1$ respectively. The sparsity optimization problem is expressed as,

$$\begin{aligned} \max_h \quad & \mathrm{Hoyer}(p_t), \\ \text{s.t.} \quad & p \ge 0,\ pJ_{n,1} = 1. \end{aligned} \tag{15}$$

Theorem 1. *Optimization problem in Eq. 15 could be precisely solved using the Karush-Kuhn-Tucker (KKT) conditions [66]. And the time complexity is $\mathcal{O}(2^n)$.*

The detailed optimization process in **Theorem 1** is as follows: Since the sum of p is 1, the optimization problem could be simplified as

$$\begin{aligned} \max_h \quad & pp^\top, \\ \text{s.t.} \quad & p \ge 0,\ J_{1,n}p = 1, \end{aligned} \tag{16}$$

where $p^\top = (Z^\top)^\dagger S_t^\top + (I_n - (Z^\top)^\dagger Z^\top)h$.

For simplicity, let $b = (Z^\top)^\dagger S_t^\top$, $A = I_n - (Z^\top)^\dagger Z^\top$. The standard form of the problem could be written as

$$\begin{aligned} \min_h \quad & -b^\top b - h^\top Ah, \\ \text{s.t.} \quad & -b - Ah \le 0,\ J_{1,n}(b + Ah) = 1. \end{aligned} \tag{17}$$

The Lagrange function is defined as

$$\begin{aligned} L(h, \lambda, \mu) = & -b^\top b - h^\top Ah + \lambda(1 - J_{1,n}(b + Ah)) \\ & + \mu(-b - Ah). \end{aligned} \tag{18}$$

Then the KKT conditions are

$$\begin{aligned} & \nabla_h L = -2Ah - \lambda A J_{n,1} - A\mu = 0, \\ & 1 - J_{1,n}(b + Ah) = 0, \\ & -b - Ah \le 0, \\ & \mu \ge 0, \\ & \mu_{diag}(-b - Ah) = 0, \end{aligned} \tag{19}$$

where $\mu_{diag} = \begin{pmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_n \end{pmatrix}$. Let $b = (b_1, \cdots, b_n)$, $A = \begin{pmatrix} A_1 \\ \vdots \\ A_n \end{pmatrix}$. We have

$$\begin{aligned} \mu_i(b_i + A_i h) = 0, \quad i = 1, \cdots, n \\ A_i(2h + \mu + \lambda J_{n,1}) = 0, \quad i = 1, \cdots, n \\ \sum_{i=1}^{n} A_i h + \sum_{i=1}^{n} b_i - 1 = 0. \end{aligned} \tag{20}$$

Suppose there are k non-zero elements in μ, indexes as $\mathfrak{N} = \{n_1, n_2, \cdots, n_k\}$. Let $\alpha_i = \text{sum}(A_i)$, $\alpha = \text{sum}(A)$. We have

$$\begin{aligned} 2b_{n_i} - A_{n_i}\mu - \lambda\alpha_{n_i} = 0, \\ \lambda\alpha = 2(\sum_{i=1}^{n} b_i - 1 - \frac{1}{2}\sum_{i=1}^{k} \mu_{n_i}\alpha_{n_i}). \end{aligned} \tag{21}$$

Further, simplify them into a form that only involves the non-zero terms

$$\begin{aligned} b_{\mathfrak{N}} = \frac{1}{2}(A_{\mathfrak{N}\mathfrak{N}}\mu_{\mathfrak{N}} + \lambda\alpha_{\mathfrak{N}}), \\ \lambda = \frac{2}{\alpha}(J_{1,n}b - 1 - \frac{1}{2}\alpha_{\mathfrak{N}}^{\top}\mu_{\mathfrak{N}}). \end{aligned} \tag{22}$$

Substitute λ into $b_{\mathfrak{N}}$

$$\frac{1}{2}(A_{\mathfrak{N}\mathfrak{N}} - \frac{1}{\alpha}\alpha_{\mathfrak{N}}\alpha_{\mathfrak{N}}^{\top})\mu_{\mathfrak{N}} = b_{\mathfrak{N}} - \frac{J_{1,n}b - 1}{\alpha}\alpha_{\mathfrak{N}}. \tag{23}$$

Then we can obtain μ, λ, h sequentially. Substitute the results into the inequality constraints of the KTT conditions and verify. If the constraints are satisfied, we fortunately find the solution. Note that we have to decide some elements of μ to zero each time. In the worst case, we need to try 2^n times.

Note in Eq. 1, we have to compute p_t at each integration step t, leading to unacceptably high time consumption. In addition, Eq. 15 could be approximately solved using iterative methods such as gradient descent. However, the time complexity is still intolerable. Therefore, we relax the conditions to allow for negative values.

Theorem 2. *By introducing negative probability, the optimization problem turns into Eq. 24, and could be precisely solved by Lagrange multipliers. The time complexity could be reduced from $\mathcal{O}(2^n)$ to $\mathcal{O}(n)$.*

By relaxing the conditions to allow for negative values, the sparsity optimization problem turns into,

$$\begin{aligned} \max_{h} \quad & \text{Hoyer}(p_t), \\ \text{s.t.} \quad & pJ_{n,1} = 1. \end{aligned} \tag{24}$$

The new problem can be solved precisely using Lagrange multipliers. The detailed optimization process in **Theorem 2** is as follows. The optimization problem is given by:

$$\begin{aligned} \min_{h} \quad & -b^{\top}b - h^{\top}Ah, \\ \text{s.t.} \quad & J_{1,n}(b + Ah) = 1. \end{aligned} \tag{25}$$

The Lagrange function is defined as

$$L(h, \lambda) = -b^{\top}b - h^{\top}Ah + \lambda(J_{1,n}(b + Ah) - 1). \tag{26}$$

Let derivatives equal 0,

$$\begin{aligned} \nabla_h L = -2Ah + \lambda(J_{1,n}A)^{\top} = 0, \\ \nabla_\lambda L = J_{1,n}(b + Ah) - 1 = 0. \end{aligned} \tag{27}$$

Noting that $A = A^{\top}$, we have

$$2Ah = \lambda A J_{n,1}. \tag{28}$$

Substituting it into the second equation, we have

$$\lambda = \frac{2 - 2J_{1,n}b}{J_{1,n}AJ_{n,1}}. \tag{29}$$

Then,

$$Ah = \frac{(1 - J_{1,n}b)AJ_{n,1}}{J_{1,n}AJ_{n,1}}. \tag{30}$$

Finally, we obtain p as

$$p^{\top} = b - \frac{(J_{1,n}b - 1)AJ_{n,1}}{J_{1,n}AJ_{n,1}}. \tag{31}$$

The most time-consuming part is the matrix summation of A, which can be computed in $\mathcal{O}(n)$ time on modern GPUs optimized for matrix operations. Therefore, the final result of the above optimization problem is,

$$p_t^{\top} = b_p - \frac{(J_{1,n}b_p - 1)A_p J_{n,1}}{J_{1,n}A_p J_{n,1}}, \tag{32}$$

where $b_p = (Z^{\top})^{\dagger}S_t^{\top}$ and $A_p = I_n - (Z^{\top})^{\dagger}Z^{\top}$.

Next, we describe how to express z_t as a function of S_t. As softmax is too complex to be directly given an algebraic expression, we perform a first-order Taylor expansion for it,

$$p_t = \frac{a + J_{1,n}}{(a + J_{1,n})J_{n,1}}. \tag{33}$$

Combining Eq. 5, Eq. 32 and Eq. 33, we have

$$\begin{aligned} z_t = \sqrt{d} \cdot a_h(Z^{\top})^{\dagger}, \\ a_h = h_2^{\top}(I_n - (J_{n,1}p - I_n)(J_{n,1}p - I_n)^{\dagger}) - J_{1,n}, \end{aligned} \tag{34}$$

where h_2 is a random vector and could be trained together with the neural network.

Finally, we apply Eq. 32 and Eq. 34 to Eq. 12, then obtain the differential equation of DHS S_t.

D. Downstream Output

DHS provides a continuous hidden embedding, which could be conveniently used for downstream tasks. Following the conventions of NODE-based methods, one straightforward approach is to directly map DHS to the desired output using a simple neural network, that is

$$y = f_{out}(S). \tag{35}$$

In the classification tasks, y refers to the label of the time series, and S refers to DHS at all integration time points. In the interpolation and extrapolation tasks, y_t at any given time is obtained from the corresponding S_t at the same time point.

DHS can also be easily combined with other methods. HiPPO [31] is an effective representation for time series and is able to update through integration. However, HiPPO requires a continuous sequence as input, which is exactly what DHS offers. We construct the following system of equations:

$$\begin{aligned}\frac{dr_t}{dt} &= f_r(S_t||c_t||r_t),\\ \frac{dc_t}{dt} &= Ac_t + B(W_r r_t),\\ \frac{dS_t}{dt} &= F_s(S_t, X_{ob}, t),\end{aligned} \tag{36}$$

where c_t is the HiPPO representation. The information is concentrated on r_t and then output through a simple neural network similar to Eq. 35.

IV. Experiments

In this section, we present extensive experiments conducted on both synthetic and real-world datasets to evaluate the proposed DIFFODE across mainstream irregular time series analysis tasks, including classification, interpolation, and extrapolation. The comprehensive experiments aim to assess DIFFODE from multiple perspectives and answer the following ***R***esearch ***Q***uestions (***RQ***):

RQ1: Can DIFFODE achieve superior accuracy in irregular time series classification compared to advanced approaches, particularly ODE-based methods? Refer to Section IV-B.

RQ2: How well does DIFFODE perform in interpolation, and extrapolation tasks compared to competing approaches, particularly ODE-based methods? Refer to Section IV-C.

RQ3: How efficient is DIFFODE relative to the baseline methods? Refer to Section IV-D.

RQ4: How about the scalability of DIFFODE when the given datasets with different scales? Refer to Section IV-E.

RQ5: What role does the Hoyer Metric play in DIFFODE? Refer to Section IV-F.

RQ6: Does each proposed component of DIFFODE contribute to the model's performance? Refer to Section IV-G.

A. Experimental Settings

1) Datasets: We implement our approach on four datasets, namely **synthetic periodic dataset**, **dynamical systems**, **USHCN**, and **PhysioNet**, whose details are as follows:

- **Synthetic periodic dataset** [42] is generated using the algebraic equation $x(t) = \sin(t+\phi)*\cos(3*(t+\phi))$ with time $t \in (0, 10)$ and phase $\phi \sim N(0, 2\pi)$. We simulate 1000 time series and create a binary label $y = \mathrm{I}(\mathrm{x}(5) > 0.5)$. To make the time series irregular, we sample from them according to a Poisson process with a rate of 70%. The dataset is divided into training, testing, and validation sets with the ratio of $50\% : 25\% : 25\%$.
- **Dynamical systems** [42] are a widely studied type of time series that require models to learn the underlying dynamics of the processes. We consider one of the representations of the most complex dynamical systems, chaotic attractors. Chaotic attractors are sensitive to initial conditions and small noises might result in exponentially diverging trajectories. We construct **Lorenz63** and **Lorenz96** systems and remove the last dimension to make it never fully observed. To make it more irregular, we further sample them using a Poisson process with a rate of 30%. Similarly, the dataset is divided into training, testing, and validation sets with the ratio of $50\% : 25\% : 25\%$.
- **The United States Historical Climatology Network (USHCN)** [67] contains over 150 years of daily climate data from the United States, including five different variables (precipitation, snowfall, snow depth, minimum and maximum temperature) from 1218 weather stations. Following the preprocessing procedure of GRU-ODE-Bayes, we select the data of 1168 stations over 4 years. Due to equipment failure or the occasional collection of certain metrics (*e.g.* snow depth), the dataset is very sparse. We further increase the irregularity by removing half of the time points and randomly removing 20% of the observations. Divide the dataset into 60% for training, 20% for testing, and 20% for validation.
- **PhysioNet Challenge 2012 (PhysioNet)** [68] includes the physical conditions of 8000 patients in the ICU during the first 48 hours, including 37 different indicators, such as serum glucose, heart rate, platelets, etc. Following the preprocessing procedure in ODE-RNN [35], we round the observations to 6 minutes. Divide the dataset into 60% for training, 20% for testing, and 20% for validation.
- **LargeST** [69] is a large dataset which originally containing 5-year traffic flow data collected by 8600 sensors with interval of 5 minutes. We aggregated the raw data at hourly intervals, resulting in a sequence with a length of $43,824$. To introduce irregularity, we randomly masked half of the data points. Divide the dataset into 60% for training, 20% for testing, and 20% for validation.

The statistics for all datasets used in the experiments are provided in Table II. These datasets were selected to address various challenges in irregular time series analysis. The synthetic periodic dataset is used for classification, offering controlled data with irregular sampling. The Lorenz63 and Lorenz96 chaotic systems test the model on dynamic, irregular data, helping evaluate how well the DHS mechanism handles nonlinear dynamics and long-term dependencies. The USHCN dataset, with 150 years of daily climate data, is used for interpolation tasks, predicting missing climate data. It is also chosen for efficiency and scalability analysis due to its large size and real-world relevance, containing high-dimensional, irregularly sampled data with missing values, making it ideal for testing DIFFODE's performance on large, real-world datasets. The PhysioNet dataset is used for extrapolation tasks to assess long-term forecasting with missing or inconsistent data. Finally, the synthetic dataset was included in the ablation study to control factors like sampling rate and noise, allowing for a controlled evaluation of the DHS mechanism. Each dataset focuses on different aspects of handling irregular time series data, from idealized settings to real-world challenges.

TABLE II: Statistics of the datasets invovled in experiments.

Dataset	Type	# of time series	Sequence length	Features/variables	Irregularity
Synthetic	Synthetic	1,000	10	1 (binary label)	70% Poisson-sampled
Lorenz63	Dynamical system	1	1,000	63	30% Poisson-sampled
Lorenz96	Dynamical system	1	1,000	96	30% Poisson-sampled
USHCN	Climate	1,168	1,461 (4 years)	5 (precipitation, snowfall, etc.)	20% random missing
PhysioNet	Hospital	8,000	48	37 (various health metrics)	Regularly sampled (6 min intervals)
LargeST	Traffic	8,600	43,824 (5 years)	1 (traffic flow)	Regularly sampled (60 min intervals)

TABLE III: Classification performance on the synthetic dataset and dynamical systems (Top-1 accuracy is reported). The **Bold** and Underline values denote the best and second-best performance, respectively, hereinafter the same.

Model	Category	Synthetic	Lorenz63	Lorenz96
mTAN	Attention-based	0.757 ± 0.030	0.718 ± 0.066	0.713 ± 0.072
ContiFormer		0.992 ± 0.006	0.989 ± 0.004	$\underline{0.987 \pm 0.004}$
HiPPO-obs	SSM-based	0.758 ± 0.023	0.837 ± 0.034	0.949 ± 0.007
HiPPO-RNN		0.742 ± 0.008	0.804 ± 0.023	0.944 ± 0.008
S4		0.994 ± 0.003	0.911 ± 0.005	0.948 ± 0.016
GRU	RNN-based	0.848 ± 0.044	0.805 ± 0.017	0.834 ± 0.058
GRU-D		0.897 ± 0.028	0.859 ± 0.015	0.864 ± 0.048
ODE-RNN	ODE-based	0.870 ± 0.032	0.813 ± 0.013	0.954 ± 0.012
Latent ODE		0.782 ± 0.014	0.713 ± 0.021	0.762 ± 0.024
GRU-ODE-Bayes		0.968 ± 0.004	0.825 ± 0.031	0.925 ± 0.004
NRDE		0.773 ± 0.111	0.604 ± 0.046	0.606 ± 0.112
PolyODE		$\underline{0.994 \pm 0.003}$	$\underline{0.992 \pm 0.000}$	0.984 ± 0.002
DIFFODE (ours)	**ODE-based**	$\mathbf{0.997 \pm 0.001}$	$\mathbf{0.993 \pm 0.001}$	$\mathbf{0.991 \pm 0.003}$

2) Baselines: We compare the performance of DIFFODE with a variety of baselines across four categories, including **attention**-based model (mTAN [70], ContiFormer [44]), **SSM**-based models (HiPPO-RNN [31], HiPPO-obs, S4 [32]), **RNN**-based models (GRU [25], GRU-D [13]), and **ODE**-based models (Latent ODE [33], ODE-RNN [35], GRU-ODE-Bayes [38], NRDE [63], PolyODE [42]). The detailed descriptions of the baselines are as follows,

i) Attention-based methods:

- **mTAN** [70] is a generative method based on a variational auto-encoder and uses attention mechanism to produce a fixed-length representation for time series of arbitrary length.
- **ContiFormer** [44] extends the relation modeling of vanilla Transformer to the continuous-time domain, which explicitly incorporates the modeling abilities of continuous dynamics of Neural ODEs with the attention mechanism of Transformers.

ii) SSM-based methods:

- **HiPPO-RNN** [31] is a recurrent neural network architecture that uses orthogonal polynomial projections of the hidden process. We also use a variant of this approach where we use the HiPPO operator directly on the observed time series, rather than on the hidden process. Following [42], we call this variant **HiPPO-obs**.
- **S4** [32] is a sequence model that uses a new parameterization for the state space model's continuous-time, recurrent, and convolutional views to efficiently model long-range dependencies in a principled manner.

iii) RNN-based methods:

- **GRU** [25] is a type of recurrent neural network architecture designed to efficiently model sequential data while mitigating issues like the vanishing gradient problem.
- **GRU-D** [13] is based on GRU, which takes two representations of missing patterns, *i.e.*, masking and time interval, and effectively incorporates them into a deep model architecture so that it not only captures the long-term temporal dependencies in time series, but also utilizes the missing patterns to achieve better prediction results.

iv) ODE-based methods:

- **Latent ODE** [33] extends Neural ODEs to sequential data by embedding observations into a latent space using an encoder. The dynamics in this latent space are modeled with an ODE, and the states are decoded back to the observation space.
- **ODE-RNN** [35] generalizes RNNs to have continuous-time hidden dynamics defined by ordinary differential equations, which provides relatively interpretable latent states, as well as explicit uncertainty estimates about latent states.
- **GRU-ODE-Bayes** [38] combines GRU with ODEs to address irregularly sampled multivariate time series, which encodes a continuity prior for the latent process and can exactly represent the Fokker-Planck dynamics of

complex processes driven by a multidimensional stochastic differential equation.

- **NRDE** [63] combines ideas from rough path theory and neural controlled differential equations to improve performance, which uses the log-signature transform of time series data and allows the model to encode long-term dependencies more efficiently.
- **PolyODE** [42] models the latent continuous-time process as a projection onto a basis of orthogonal polynomials. This formulation enforces long-range memory and preserves a global representation of the underlying dynamical system.

We adopt the configurations that yield the best performance for each baseline to run their official codes on the same machine used for running our model. Traditional attention-based models often struggle with irregular time series due to their reliance on fixed-length representations and local dependencies, which can fail to capture the continuity of temporal dynamics. In contrast, our Differentiable Hidden State (DHS) mechanism addresses this limitation by using irregularly sampled observations as Key and Value matrices, allowing for dynamic updates of the latent state and better capturing long-range dependencies. By incorporating generalized inverses, the DHS mechanism ensures context-aware attention computation, preserving continuity and enabling more accurate modeling of temporal relationships. This approach circumvents the issues of fragmented representation seen in traditional models, making it particularly effective for tasks like interpolation, extrapolation, and long-term forecasting with irregular data.

3) Evaluation Metrics: To rigorously evaluate the performance of the proposed model across classification, interpolation, and extrapolation, we adopt task-specific evaluation metrics that comprehensively reflect the effectiveness of DIFFODE in capturing the underlying dynamics of irregular time series. The detailed descriptions of the evaluation metrics involved are as follows:

Top-1 Accuracy for Classification. The classification task is assessed using the Top-1 accuracy metric, which quantifies the proportion of test samples for which the predicted class matches the ground truth label and is formally defined as:

$$\text{Top-1 Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = y_i), \tag{37}$$

where N denotes the total number of test samples, $\hat{y}_i$ represents the predicted class of the i-th sample, y_i is the corresponding ground truth, and $\mathbb{1}(\cdot)$ is the indicator function. A higher Top-1 accuracy signifies superior classification performance.

Mean Squared Error for Interpolation and Extrapolation. For interpolation and extrapolation tasks, we use Mean Squared Error (MSE) to evaluate performance. It measures the average squared differences between predicted and actual values and is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}_{t_i} - y_{t_i})^2 \times 10^{-2}, \tag{38}$$

where N is the number of evaluation points, $\hat{y}t_i$ is the predicted value at time t_i, and yt_i is the ground truth value. Lower MSE values indicate better accuracy in predicting and reconstructing irregular time series.

4) Implementation Details: The proposed DIFFODE framework involves three small neural networks: (1) the input mapping from observations to latent states, (2) the dynamics modeling of the DHS, and (3) the output mapping. A one-layer GRU is used to map observations into latent states, while an MLP with one hidden layer models the dynamics of the DHS. The output mapping is also implemented using an MLP with one hidden layer. For all datasets, the hidden size of the MLPs is set to 32. The ODE integration is performed using the implicit Adams method, an adaptive numerical integration method known for its tiny numerical errors. Early stopping is applied if the validation loss does not improve for 20 consecutive epochs. The learning rate and weight decay are both set to 0.001.

For **classification** tasks, the batch size is set to 128, and the dimension of DHS and information state r_t is set to 16. The integration step of the ODE solution is set to 0.05. When we train the model, we have 250 max epochs. For **interpolation** and **extrapolation** tasks, the batch size is set to 32. The dimension of DHS and information state r_t is set to 32. The integration step of the ODE solution is set to 5. The maximum number of training epochs is set to 100.

B. Performance Comparison of Classification (RQ1)

Classification is an important application of irregular time series analysis. In our evaluation, we subjected a variety of models to rigorous testing using both synthetic periodic datasets and dynamical systems, employing cross-entropy loss for training purposes. The results, presented in Table III, reveal that our proposed model, DIFFODE, surpasses a diverse array of existing methods, achieving state-of-the-art performance across all tested datasets. Notably, the attention-based method mTAN, along with the RNN-based methods GRU and GRU-D, were unable to surpass other approaches. This underperformance is attributed to their discrete frameworks, underscoring the significant advantage offered by our model's continuous hidden state representation. While the recent PolyODE model demonstrates a general capacity to extract temporal information from time series, it falls short in accurately capturing the subtleties of the underlying dynamics when compared to the robust capabilities of our proposed DIFFODE.

C. Performance Comparison of Interpolation and Extrapolation (RQ2)

We employ the USHCN, PhysioNet, and LargeST datasets [69] to evaluate the performance of models on interpolation and extrapolation tasks. For interpolation, our goal is to reconstruct the complete time series from a subset of available observations. Conversely, in the extrapolation task, we divide the time series into two equal parts: the first half is utilized for model training, while the full sequence is employed for making predictions. The results, as detailed in

TABLE IV: Interpolation and extrapolation performance on USHCN, PhysioNet, and LargeST (MSE is reported).

Model	USHCN		PhysioNet		LargeST	
	Interpolation	*Extrapolation*	*Interpolation*	*Extrapolation*	*Interpolation*	*Extrapolation*
mTAN	1.766 ± 0.009	2.360 ± 0.038	0.208 ± 0.025	0.340 ± 0.020	411.81 ± 63.23	466.58 ± 67.34
ContiFormer	0.837 ± 0.057	1.634 ± 0.082	0.212 ± 0.023	0.376 ± 0.034	413.62 ± 42.19	457.52 ± 53.82
HiPPO-obs	1.268 ± 0.051	2.417 ± 0.068	0.323 ± 0.061	0.855 ± 0.024	475.82 ± 63.58	522.62 ± 51.85
HiPPO-RNN	1.172 ± 0.061	2.324 ± 0.031	0.293 ± 0.068	0.769 ± 0.053	457.25 ± 72.25	497.25 ± 72.10
S4	0.823 ± 0.016	1.504 ± 0.063	0.229 ± 0.023	0.535 ± 0.067	437.73 ± 73.34	453.73 ± 64.99
GRU	1.068 ± 0.073	2.071 ± 0.015	0.364 ± 0.088	0.880 ± 0.140	522.36 ± 74.43	522.36 ± 67.71
GRU-D	0.994 ± 0.011	1.718 ± 0.015	0.338 ± 0.027	0.873 ± 0.071	524.13 ± 6.84	527.46 ± 54.87
ODE-RNN	0.831 ± 0.008	1.955 ± 0.467	0.236 ± 0.009	0.467 ± 0.006	417.45 ± 24.54	451.15 ± 54.62
Latent ODE	1.798 ± 0.009	2.034 ± 0.005	0.212 ± 0.027	0.725 ± 0.072	467.26 ± 73.12	527.18 ± 64.83
GRU-ODE-Bayes	0.841 ± 0.142	5.437 ± 1.020	0.521 ± 0.038	0.798 ± 0.071	486.82 ± 68.28	513.42 ± 54.81
NRDE	0.961 ± 0.051	1.923 ± 0.607	0.434 ± 0.077	0.819 ± 0.037	517.35 ± 64.24	557.95 ± 64.93
PolyODE	0.806 ± 0.017	1.842 ± 0.440	0.205 ± 0.041	0.598 ± 0.034	425.63 ± 53.62	485.57 ± 52.45
DIFFODE (ours)	$\mathbf{0.765 \pm 0.023}$	$\mathbf{0.869 \pm 0.043}$	$\mathbf{0.175 \pm 0.074}$	$\mathbf{0.308 \pm 0.054}$	$\mathbf{365.14 \pm 37.45}$	$\mathbf{396.23 \pm 34.10}$

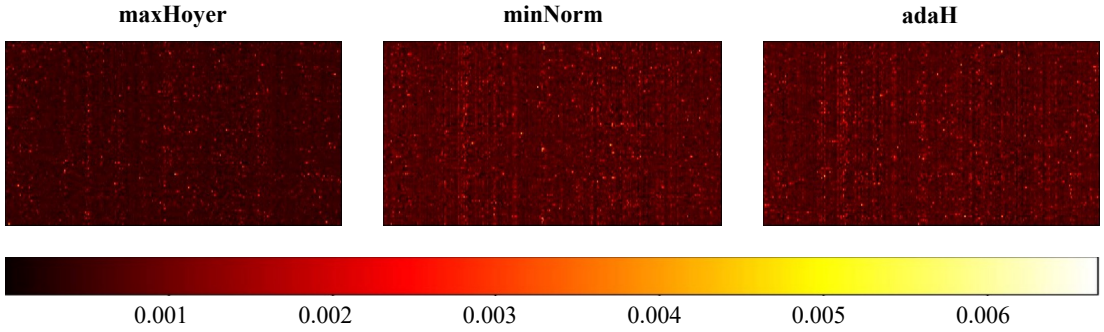


Fig. 3: Visualization of attention scores obtained using different methods for determining p_t. A lower number of lighter-colored points indicates greater sparsity, highlighting the differences in sparsity levels across the methods.

Table IV and expressed in terms of mean squared error (MSE) scaled by a factor of 10^{-2}, demonstrate our proposed model, DIFFODE, outperforming alternative methods, particularly in the extrapolation task. Specifically, DIFFODE surpasses the best results of all baselines by $\{5.1\%, 14.6\%, 11.7\%\}$ on the interpolation tasks for the USHCN and PhysioNet datasets, respectively. Moreover, DIFFODE achieves even more remarkable performance improvements over the best-performing baselines on extrapolation, with ratios of 42.2%, 9.4%, and 12.2% on USHCN, PhysioNet, and LargeST, respectively. This superior performance indicates that DIFFODE is proficient at capturing the intrinsic dynamics of time series data, a capability that substantially enhances its ability to accurately forecast future trends. Therefore, in the three tasks of classification, interpolation, and extrapolation on irregular time series, failing to fully leverage the contextual information in the data can result in several issues, including inaccurate forecasting, fragmented representations, loss of temporal continuity, and poor generalization. When models do not capture the full context, they may overlook important long-range dependencies, leading to less reliable predictions and overfitting to local patterns. Our DHS mechanism addresses these challenges by enabling context-aware attention and preserving the continuity of latent dynamics, thereby improving the handling of irregular data and enhancing overall model performance.

TABLE V: Comparison analysis of model efficiency.

Model	Complexity	Time (*s/epoch*)
ContiFormer	$\mathcal{O}(d^2n^2L)$	154
HiPPO-obs	$\mathcal{O}(d_c^2L)$	86
GRU-D	$\mathcal{O}(d^2n)$	232
ODE-RNN	$\mathcal{O}(d^2L)$	91
Latent ODE	$\mathcal{O}(d^2L)$	110
PolyODE	$\mathcal{O}(d_c^2d^2L)$	131
DIFFODE (ours)	$\boldsymbol{\mathcal{O}(d_c^2nL)}$	**126**

D. Efficiency Analysis (RQ3)

In this part, we theoretically calculate the time complexity of the proposed DIFFODE, and then compare the training time consumption of DIFFODE with that of baselines. We compare the time complexity of our method with representative baselines as presented in Table V. According to the table, we

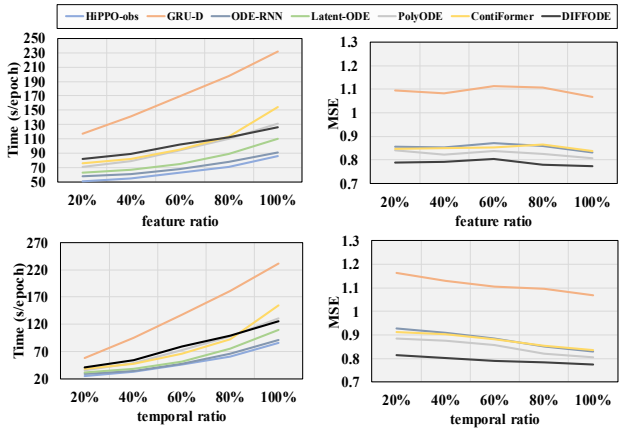


Fig. 4: Scalability analysis of DIFFODE across two key dimensions of time series data: number of features and time steps.

have n denoting the number of time points with observations, d denoting the dimension of the feature of observations, d_c denoting the dimension of the HiPPO matrix, and L denoting the integration steps. The scale of d_c is typically similar to that of d, and L is always less than n. SSM-based models, *e.g.*, HiPPO-obs, are efficient linear models and usually need at least $\mathcal{O}(d_c^2 L)$ time. RNN-based models are simple but less efficient, which usually need only $\mathcal{O}(d^2 n)$ time. ODE-based models need at least $\mathcal{O}(d^2 L)$ time, and extra time consumption related to the specific design of the model. Methods combining attention mechanism with NODE usually need $\mathcal{O}(d^2 n^2 L)$ time consumption. Our model designs reduce it to $\mathcal{O}(d^2 n L)$, with a similar time complexity as normal attention-based models. We can find that, our model achieves impressive performance gain by introducing a continuous attention mechanism while requiring acceptable additional time consumption. The time consumption of our model and baselines in one training epoch on the USHCN dataset is also listed in Table V. From the table, we can find that theoretically faster models do not necessarily have shorter training time, e.g., stacking building blocks does not increase time complexity but leads to longer training time. Therefore, to further evaluate the time consumption of DIFFODE, we next train our model on datasets with different scales and observe the time consumption differences of our model on the datasets.

E. Scalability Analysis (RQ4)

We further evaluate the scalability of our method when fed with datasets with different scales, *i.e.*, datasets with different temporal lengths and with different numbers of features. To this end, we extract a series of subdatasets of USHCN. As mentioned above, the used USHCN dataset contains 4-year data of 1168 stations, we extract $\{20\%, 40\%, 60\%, 80\%, 100\%\}$ of the stations to construct feature-wise sub-datasets of USHCN, resulting in five datasets with $\{234, 467, 701, 934, 1168\}$ stations respectively. Similarly, 5 temporal-wise subdatasets are extracted, where the temporal lengths of the subdatasets are $\{20\%, 40\%, 60\%, 80\%, 100\%\}$ of origin USHCN. The proposed DIFFODE is compared with six well-performed baselines on the interpolation task. We compare the training time consumption and the interpolation performance of the models.

Fig. 4 presents the result. It can be observed that as the dataset scale escalates, whether in terms of feature count or temporal length, the training time consumption of DIFFODE increases at a slower rate compared to all other baselines. This observation aligns with our analysis of model complexity, confirming that DIFFODE outperforms other baselines in efficiency when dealing with larger datasets. Additionally, we observe that all models exhibit robustness to changes in feature scale. This outcome is attributed to the fact that in the analysis of irregular time series, feature correlations are often limited or difficult to extract, and the difficulty of analyzing a larger number of features is comparable to that of analyzing a smaller number of features. When the available data length diminishes, the performance of all models declines accordingly. However, it is noteworthy that the proposed DIFFODE experiences the least interpolation performance degradation. The robustness of DIFFODE is demonstrated to surpass that of the baselines. In summary, this section illustrates that DIFFODE surpasses baselines in managing large-scale data and demonstrates greater robustness in the face of varying data availability.

F. Effect of Hoyer Metric (RQ5)

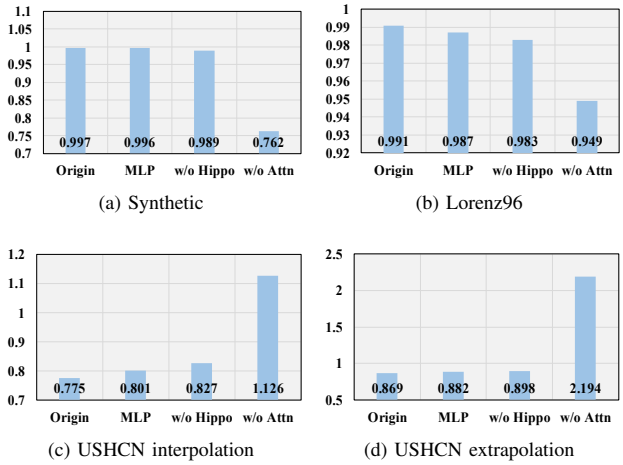
To assess the impact of maximizing the Hoyer metric (**maxHoyer**) on model performance, we conducted a comparative analysis with two alternative approaches for determining p_t: one that employs p_t with the minimum norm (**minNorm**), and another that treats h in Eq. 13 as an adaptable parameter co-trained with the neural network (**adaH**). Fig. 3 illustrates the gray-scale maps of p_t as derived from these various methods, while Table VI presents the mean squared error (MSE), scaled by 10^{-2}.

The results indicate that p_t obtained through the maximization of the Hoyer metric not only exhibits greater sparsity but also delivers superior performance on the dataset. This finding emphasizes the Hoyer metric's efficacy in promoting sparsity, which in turn is beneficial for capturing the complex interdependencies among highly correlated points within a time series.

Interestingly, the performance of p_t derived from both the minimum norm approach (minNorm) and the adaptive parameter training (adaH) is quite comparable. This similarity in performance might stem from the necessity for h to be closely aligned with the data characteristics for each batch. If h is not well-correlated with the data, its capacity to absorb meaningful information is constrained. The p_t resulting from the Hoyer metric maximization is inherently connected to S_t, aligning with the requirement for data-sensitive h values and thus explaining its enhanced performance.

TABLE VI: Performance of DIFFODE with different methods for calculating p_t on the USHCN and PhysioNet datasets. maxH refers to p_t calculated by maximizing the Hoyer metric, minN refers to p_t obtained by minimizing the norm, and trainP refers to p_t where the parameter is learned during training.

Model		maxHoyer	minNorm	adaH
USHCN	***Interpolation***	**0.765 ± 0.023**	0.804 ± 0.020	0.798 ± 0.038
	Extrapolation	**0.869 ± 0.043**	0.922 ± 0.034	0.913 ± 0.081
PhysioNet	***Interpolation***	**0.175 ± 0.074**	0.201 ± 0.076	0.197 ± 0.094
	Extrapolation	**0.308 ± 0.054**	0.346 ± 0.049	0.351 ± 0.063



(a) Synthetic (b) Lorenz96

(c) USHCN interpolation (d) USHCN extrapolation

Fig. 5: Ablation study of input neural network and output mechanism. Synthetic, Lorenz96, and USHCN datasets are employed here.

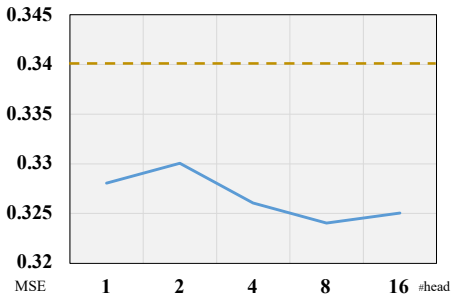


Fig. 6: Extrapolation performance on PhysioNet with different numbers of heads in attention.

G. Ablation Study (RQ6)

We come up with three more ablation studies on the input neural network, the output mechanism, and multi-head attention in this section. For the input neural network, we compare the performance of GRU and MLP on dynamical systems. When using MLP, we actually have $E(x_t)$ in Eq. 4 as $\varnothing$. For the output mechanism, we compare the performance of using and not using the HiPPO mechanism. The result is shown in Fig. 5. Synthetic, Lorenz96, and USHCN are employed here. It is shown that using GRU as an input layer could better capture the information over time, and HiPPO is even more important in generating predictions. For multi-head attention, we first remove the attention and evaluate the impact of removing the attention mechanism. When removing attention, the model architecture is similar to HiPPO-RNN. As shown in Fig. 5, the variant *w/o Attn* performs far worse than original model. We then compare the performance of the model with different heads on the PhysioNet dataset. The result is shown in Fig. 6, which illustrates that the improvement from multi-head attention is limited, but it incurs additional time consumption overhead.

V. Conclusion

This paper tackles a significant challenge faced by current neural ODE methods: their inability to seamlessly integrate contextual information while preserving the continuity of the latent dynamics in irregular time series. To overcome this challenge, we propose a novel data-driven neural ODE framework (DIFFODE), with an innovative attention-based differential hidden state space, leveraging irregularly sampled observations as Key and Value matrices to enrich the model's context awareness. Building upon this novel hidden state space, we employ the theory of generalized inverses to formulate an ODE that encapsulates the dynamics of the hidden states over time. Furthermore, to enhance the precision of temporal relationships, we incorporate the Hoyer metric, aiming to maximize the sparsity of attention scores during the generation of hidden states. Our approach has been rigorously compared with existing state-of-the-art methods on both synthetic and real-world datasets, with experimental results consistently showcasing the superior effectiveness of our model in diverse irregular time series tasks, especially on interpolation and extrapolation. Additionally, we evaluate the scalability of our method when fed with datasets with different scales, and illustrate that DIFFODE outperforms existing works in managing datasets with different scales. Overall, our work not only provides a brand-new and effective solution for irregular time series analysis, but also paves the way for broader applications in data management.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 12227901, the Project of Stable Support for Youth Team in Basic Research Field, CAS under Grant YSBR-005, and the Natural Science Foundation of Jiangsu Province under Grant BK20240460.

REFERENCES

[1] S. K. Jensen, T. B. Pedersen, and C. Thomsen, "Time series management systems: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2581–2600, 2017.

[2] Q. Wen, L. Yang, T. Zhou, and L. Sun, "Robust time series analysis and applications: An industrial perspective," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4836–4837.

[3] Y. Zhang, X. Wang, X. Yu, Z. Sun, K. Wang, and Y. Wang, "Drawing informative gradients from sources: A one-stage transfer learning framework for cross-city spatiotemporal forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[4] C. Liu, H. Miao, Q. Xu, S. Zhou, C. Long, Y. Zhao, Z. Li, and R. Zhao, "Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation," in *41th IEEE International Conference on Data Engineering*, 2025.

[5] H. Miao, Y. Zhao, C. Guo, B. Yang, K. Zheng, and C. S. Jensen, "Spatio-temporal prediction on streaming data: A unified federated continuous learning framework," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[6] C. Wang, X. Huang, J. Qiao, T. Jiang, L. Rui, J. Zhang, R. Kang, J. Feinauer, K. A. McGrail, P. Wang *et al.*, "Apache iotdb: Time-series database for internet of things," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2901–2904, 2020.

[7] S. Bauer, B. Schölkopf, and J. Peters, "The arrow of time in multivariate time series," in *International Conference on Machine Learning*. PMLR, 2016, pp. 2043–2051.

[8] J. Jia and A. R. Benson, "Neural jump stochastic differential equations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[9] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, "Transformer hawkes process," in *International conference on machine learning*. PMLR, 2020, pp. 11 692–11 702.

[10] Y. Zhang, B. Wang, Z. Shan, Z. Zhou, and Y. Wang, "Cmt-net: A mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1406–1414.

[11] T. Pelkonen, S. Franklin, J. Teller, P. Cavallaro, Q. Huang, J. Meza, and K. Veeraraghavan, "Gorilla: A fast, scalable, in-memory time series database," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1816–1827, 2015.

[12] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 351–360.

[13] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[14] X. Chen, X. Li, B. Liu, and Z. Li, "Biased temporal convolution graph network for time series forecasting with missing values," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=O9nZCwdGcG

[15] X. Chen, X. Li, X. Chen, and Z. Li, "Structured matrix basis for multivariate time series forecasting with interpretable dynamics," *Advances in Neural Information Processing Systems*, vol. 37, pp. 24 326–24 349, 2024.

[16] F. Chen, Y. Zhang, Z. Qin, L. Fan, R. Jiang, Y. Liang, Q. Wen, and S. Deng, "Learning multi-pattern normalities in the frequency domain for efficient time series anomaly detection," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 747–760.

[17] M. Ma, J. Hu, C. S. Jensen, F. Teng, P. Han, Z. Xu, and T. Li, "Learning time-aware graph structures for spatially correlated time series forecasting," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 4435–4448.

[18] R. Zha, L. Zhang, S. Li, J. Zhou, T. Xu, H. Xiong, and E. Chen, "Scaling up multivariate time series pre-training with decoupled spatial-temporal representations," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 667–678.

[19] Y. Fang, J. Xie, Y. Zhao, L. Chen, Y. Gao, and K. Zheng, "Temporal-frequency masked autoencoders for time series anomaly detection," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1228–1241.

[20] X. Ma, X. Hong, S. Lu, and W. Li, "Ts3net: Triple decomposition with spectrum gradient for long-term time series analysis," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 887–900.

[21] X. Ding, Y. Li, H. Wang, C. Wang, Y. Liu, and J. Wang, "Tsddiscover: Discovering data dependency for time series data," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 3668–3681.

[22] H. Miao, Z. Liu, Y. Zhao, C. Guo, B. Yang, K. Zheng, and C. S. Jensen, "Less is more: Efficient time series dataset condensation via two-fold modal matching," *Proceedings of the VLDB Endowment*, vol. 18, no. 2, pp. 226–238, 2024.

[23] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," *Advances in neural information processing systems*, vol. 31, 2018.

[24] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[26] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, T. Geng, and Y. Xu, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2021.

[27] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019.

[28] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.

[29] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[30] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

[31] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," *Advances in neural information processing systems*, vol. 33, pp. 1474–1487, 2020.

[32] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=uYLFoz1vlAC

[33] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[34] M. Habiba, B. A. Pearlmutter, and M. Maleki, "Recent trends in modelling the continuous time series using deep learning: A survey," *arXiv preprint arXiv:2409.09106*, 2024.

[35] Y. Rubanova, R. T. Chen, and D. K. Duvenaud, "Latent ordinary differential equations for irregularly-sampled time series," *Advances in neural information processing systems*, vol. 32, 2019.

[36] M. Lechner and R. Hasani, "Learning long-term dependencies in irregularly-sampled time series," *arXiv preprint arXiv:2006.04418*, 2020.

[37] J.-T. Chien and Y.-H. Chen, "Learning continuous-time dynamics with attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1906–1918, 2022.

[38] E. De Brouwer, J. Simm, A. Arany, and Y. Moreau, "Gru-ode-bayes: Continuous modeling of sporadically-observed time series," *Advances in neural information processing systems*, vol. 32, 2019.

[39] C. Herrera, F. Krach, and J. Teichmann, "Neural jump ordinary differential equations: Consistent continuous-time prediction and filtering," *arXiv preprint arXiv:2006.04727*, 2020.

[40] M. Poli, S. Massaroli, J. Park, A. Yamashita, H. Asama, and J. Park, "Graph neural ordinary differential equations," *arXiv preprint arXiv:1911.07532*, 2019.

[41] J. Oskarsson, P. Sidén, and F. Lindsten, "Temporal graph neural networks for irregular data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 4515–4531.

[42] E. D. Brouwer and R. G. Krishnan, "Anamnesic neural differential equations with orthogonal polynomial projections," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=xYWqSjBcGMl

[43] P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6696–6707, 2020.

[44] Y. Chen, K. Ren, Y. Wang, Y. Fang, W. Sun, and D. Li, "Contiformer: Continuous-time transformer for irregular time series modeling," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[45] J. Li, Z. Zhu *et al.*, "Neural lad: A neural latent dynamics framework for times series modeling," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[46] N. Hurley and S. Rickard, "Comparing measures of sparsity," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4723–4741, 2009.

[47] J. C. B. Gamboa, "Deep learning for time-series analysis," *arXiv preprint arXiv:1701.01887*, 2017.

[48] X. Wang, H. Zhang, P. Wang, Y. Zhang, B. Wang, Z. Zhou, and Y. Wang, "An observed value consistent diffusion model for imputing missing values in multivariate time series," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2409–2418.

[49] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 6555–6565.

[50] Y. Zhang, P. Wang, B. Wang, X. Wang, Z. Zhao, Z. Zhou, L. Bai, and Y. Wang, "Adaptive and interactive multi-level spatio-temporal network for traffic forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[51] Y. Zhang, X. Wang, P. Wang, B. Wang, Z. Zhou, and Y. Wang, "Modeling spatio-temporal mobility across data silos via personalized federated learning," *IEEE Transactions on Mobile Computing*, 2024.

[52] Y. Zhang, X. Wang, Z. Sun, P. Wang, B. Wang, L. Li, and Y. Wang, "Meta koopman decomposition for time series forecasting under temporal distribution shifts," *Advanced Engineering Informatics*, vol. 62, p. 102840, 2024.

[53] C. Liu, S. Yang, Q. Xu, Z. Li, C. Long, Z. Li, and R. Zhao, "Spatial-temporal large language model for traffic prediction," in *25th IEEE International Conference on Mobile Data Management*, 2024, pp. 31–40.

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[55] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *arXiv preprint arXiv:2205.13504*, 2022.

[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[58] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[60] C. Liu, Q. Xu, H. Miao, S. Yang, L. Zhang, C. Long, Z. Li, and R. Zhao, "TimeCMA: Towards llm-empowered multivariate time series forecasting via cross-modality alignment," in *Thirty-Nineth AAAI Conference on Artificial Intelligence*, 2025.

[61] H. Miao, Y. Zhao, C. Guo, B. Yang, K. Zheng, F. Huang, J. Xie, and C. S. Jensen, "A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 2024, pp. 1050–1062.

[62] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of International Conference on Learning Representations*, 2021.

[63] J. Morrill, C. Salvi, P. Kidger, and J. Foster, "Neural rough differential equations for long time series," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7829–7838.

[64] E. H. Moore, "On the reciprocal of the general algebraic matrix," *Bulletin of the american mathematical society*, vol. 26, pp. 294–295, 1920.

[65] R. Penrose, "A generalized inverse for matrices," in *Mathematical proceedings of the Cambridge philosophical society*, vol. 51, no. 3. Cambridge University Press, 1955, pp. 406–413.

[66] G. Gordon and R. Tibshirani, "Karush-kuhn-tucker conditions," *Optimization*, vol. 10, no. 725/36, p. 725, 2012.

[67] M. J. Menne, C. N. Williams Jr, and R. S. Vose, "The us historical climatology network monthly temperature data, version 2," *Bulletin of the American Meteorological Society*, vol. 90, no. 7, pp. 993–1008, 2009.

[68] L. Citi and R. Barbieri, "Physionet 2012 challenge: Predicting mortality of icu patients using a cascaded svm-glm paradigm," in *2012 Computing in Cardiology*. IEEE, 2012, pp. 257–260.

[69] X. Liu, Y. Xia, Y. Liang, J. Hu, Y. Wang, L. Bai, C. Huang, Z. Liu, B. Hooi, and R. Zimmermann, "Largest: A benchmark dataset for large-scale traffic forecasting," *Advances in Neural Information Processing Systems*, vol. 36, pp. 75 354–75 371, 2023.

[70] S. N. Shukla and B. M. Marlin, "Multi-time attention networks for irregularly sampled time series," *arXiv preprint arXiv:2101.10318*, 2021.