Fault Diagnosis of Irregular Sequences by Adjoint Learning in Continuous-Time Model Space

Xiren Zhou, Chuyang Wei, Ao Chen, Shikang Liu, Xiangyu Wang, Huanhuan Chen*

School of Computer Science and Technology, University of Science and Technology of China
zhou0612@ustc.edu.cn, weichy2023@mail.ustc.edu.cn, chenao57@mail.ustc.edu.cn, skliu00@mail.ustc.edu.cn, sa312@ustc.edu.cn, hchen@ustc.edu.cn

Abstract

Fault Diagnosis (FD) on sequential data suffers from irregular sampling (with missing values), limited training data, and varying underlying environments. In response, this paper proposes FD by adjoint learning in continuous-time model space. Model-Space Learning employs well-fitted models that capture data's dynamics (i.e., changing information) as more stable and concise representations of the original data. The Continuous-Time Reservoir Computing Network (CT-Res) is first introduced, which embeds Ordinary Differential Equation (ODE) within the reservoir-based hidden layer to govern continuous-time hidden-state evolution, naturally handling irregular sampling without relying on fixed time steps and effectively capturing intrinsic data dynamics. By fitting each sequence via CT-Res and representing it with the fitted model, the original sequences are mapped from the data space into the continuous-time model space. We further develop an adjoint learning strategy by incorporating a discrete-time "adjoint Echo State Network (ESN)" that shares structure and parameters with CT-Res, thus enabling efficient training by bypassing the computationally intensive ODE solver, with joint optimization of fitting accuracy and class discrimination in the model space. Experiments on multiple FD benchmarks highlight the effectiveness and efficiency of our study, particularly with missing values and scarce training data.

Introduction

Fault Diagnosis (FD) on sequential data typically aims to determine whether a given sequence represents normal or is associated with a specific fault type. However, traditional diagnostic techniques often assume uniformly sampled data collected at regular intervals, an assumption that does not always hold in practical scenarios. For instance, sensor data from mechanical equipment may exhibit irregular sampling or missing values due to harsh measurement environments or sensor malfunctions. These irregular sequences, characterized by non-uniform time intervals between observations, pose substantial challenges for subsequent analysis.

A common approach to handling such irregularities involves resampling or interpolation (Sterne et al. 2009), but these techniques risk distorting the original signal and losing critical data-inherent changing information. Recurrent Neural Networks (RNNs) have also been applied (Che et al. 2018), though their high computational cost limits their applicability in time-sensitive tasks. More recently, continuous-time neural networks such as ODE-RNN (Rubanova, Chen, and Duvenaud 2019) and Neural Controlled Differential Equations (Neural CDEs) (Kidger et al. 2020; Jhin, Lee, and Park 2023) have shown promise for handling irregular sequences. These differential equation–based methods model continuous-time changing patterns by incorporating a numerical differential equation solver in forward propagation. Despite their flexibility, directly training these networks inevitably requires gradient backpropagation to differentiate through the numerical solver, which incurs significant training time and computational overhead. Such inefficiency severely limits their suitability in general, let alone for FD scenarios that demand rapid response and operate under limited data availability.

Given the challenges inherent in FD tasks, Model-Space Learning (MSL) offers an alternative framework (Chen et al. 2013). MSL shifts the data representation from the original data space to a model space by fitting the data with proper models that effectively capture their underlying dynamics (i.e., changing information). These fitted models serve as compact and stable representations of the original data, allowing learning or classification algorithms to operate efficiently on the models rather than raw input data. Demonstrably applied to diagnosing the Barcelona water network (Quevedo et al. 2014) and the Tennessee Eastman Process (Chen, Tiňo, and Yao 2014), as well as various time-series classification tasks (Liu, Zhou, and Chen 2025), MSL has proven effective using Echo State Networks (ESN) for data fitting and representation. Notably, MSL focuses on the data-intrinsic dynamics and typically requires less training data and lower computational cost than many deep learning methods, particularly when equipped with well-designed model configurations and dynamic capture (Ma et al. 2020).

Despite potential, existing MSL implementations typically fit data using step-by-step recurrent updates with fixed time steps. This assumes uniformly sampled sequences and limits their ability to accurately capture dynamics in irregularly sampled or partially missing data. Recent efforts have explored integrating ODEs for irregular-sequence fitting and representation (Chen, Zhou, and Chen 2025). Nevertheless,

*Corresponding Authors: Huanhuan Chen.

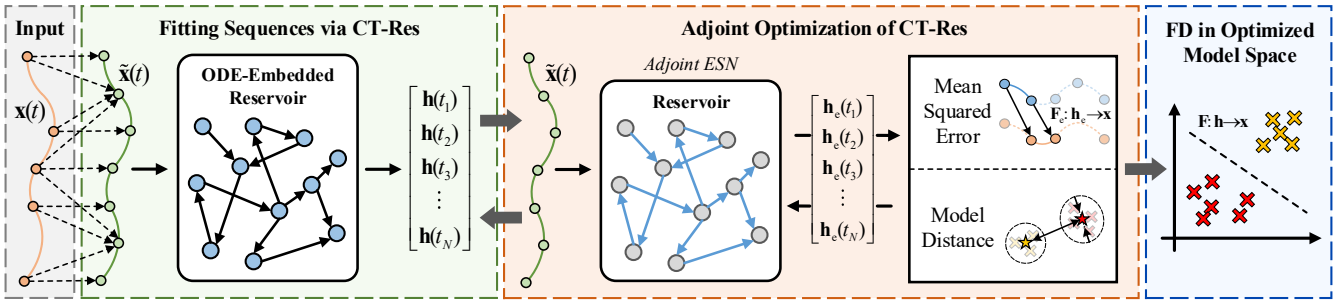


Figure 1: Overview of FD by adjoint learning in the continuous-time model space. 1) Irregular sequences are fitted using CT-Res that incorporates an ODE-embedded reservoir to capture continuous-time dynamics within each sequence, and each fitted model serves as a compact model representation of the original sequence. 2) To enable efficient optimization, a discrete-time "adjoint ESN" with shared structure and parameters is introduced, allowing rapid training based on fitting and discrimination objectives. 3) The optimized shared parameters and model distance metric are transferred back to CT-Res, enhancing its fitting accuracy and class discrimination of fitted models. FD is then performed by classifying models in the optimized model space.

using fixed-parameter networks for data fitting and dynamic capture, together with non-adjustable distance metrics to evaluate differences between models fitted from diverse sequences, remains insufficient for adapting to varying data-collecting scenarios. Such limitations pose significant challenges for accurate data fitting and representation, as well as for constructing "class-discriminative" model distributions.

Addressing the above, this study proposes FD for irregular sequences by adjoint learning in continuous-time model space, as illustrated in Fig. 1. The Continuous-Time Reservoir Computing Network (CT-Res) is first introduced. Unlike typical reservoir computing approaches that update hidden states in a discrete step-by-step manner, CT-Res integrates ODE into the reservoir-based hidden layer to govern the evolution of hidden states in continuous time, allowing the hidden state to evolve continuously under the combined influence of both recent inputs and past states, thereby preserving temporal dependencies while adapting to new observations. Such a continuous-time mechanism allows CT-Res to naturally align with non-uniform sampling and eliminates the reliance on fixed time steps. Fitting each sequence with CT-Res effectively captures the data-inherent dynamics, obtaining a compact fitted readout model[1]. Representing each sequence using a fitted model maps the original data into a continuous-time model space.

To enhance the dynamic-capture adaptability and class discrimination under varying data scenarios, we further develop an adjoint learning strategy for efficient optimization of CT-Res. Specifically, a discrete-time "adjoint ESN" is constructed, sharing the same structure, parameters, and dynamic-capture mechanism with CT-Res. It operates on interpolated inputs and updates hidden states via Euler approximation, enabling fast training without invoking the computationally intensive ODE solver. A trainable model distance metric is introduced, and two joint objectives are defined: a fitting loss for accurate dynamic capture and a discrimination loss to promote class separation in the model space. After training, the optimized parameters and metric are transferred to CT-Res, enhancing fitting accuracy and improving class discriminability to better distinguish models fitted from different types of sequences. Through the above, sequences that share similar dynamics would be mapped to similar readout models, while those containing faults yield significantly different models due to the divergent dynamics captured. Final classification is then performed within this optimized model space by distinguishing these fitted models, enabling accurate identification of corresponding sequences associated with different underlying conditions.

The main contributions of this paper are as follows:

- We introduce CT-Res, which integrates ODE into the hidden layer to continuously govern hidden-state evolution, naturally accommodating irregularly sampled sequences without reliance on fixed time steps. Fitting each sequence via CT-Res effectively captures its inherent dynamics into a compact readout model, mapping original sequences into a continuous-time model space.
- An adjoint learning strategy is proposed for efficient optimization. A discrete-time "adjoint ESN", sharing structure and parameters with CT-Res, is introduced for rapid training by bypassing the time-consuming ODE solver. Joint fitting and discrimination objectives enable adaptive optimization, enhancing CT-Res's fitting accuracy and class discriminability in the model space.
- We validate our approach across multiple FD benchmarks with irregular sequences. The results consistently highlight its effectiveness under limited training data and its improved training efficiency compared to baselines.

Related Work

Irregular-Sequence Analyzing

Most sequence analysis assumes uniformly sampled data, yet real-world sequences potentially exhibit irregular intervals due to inconsistent measurements or missing values (Chowdhury et al. 2023), which limits their applicability.

[1]In this study, "CT-Res" refers to the network designed to fit irregular sequences, yielding the "CT-Res fitted readout model" for data representation, also shortened as the "fitted readout model", "CT-Res model", "readout model", or "fitted model" for brevity.

A variety of methods have been proposed to address sequence irregularity. GRU-D (Che et al. 2018) extends GRU with decay and masking mechanisms to handle missing data. Transformer-based approaches like mTAN (Shukla and Marlin 2021) and Warpformer (Zhang et al. 2023a) adopt time-aware attention for non-uniform sampling. An alternative direction involves continuous-time analysis via differential equations. ODE-RNN (Rubanova, Chen, and Duvenaud 2019) integrates ODE solvers into recurrent networks, while Neural CDE (Kidger et al. 2020) generalizes this framework to learn from continuous input paths, building on Neural ODE (Chen et al. 2018). Besides, ODE-RSSM (Yuan et al. 2023) further applies these ideas in a recurrent state-space formulation. Although these methods exhibit representational capacity for irregular sequences, their reliance on heavily parameterized architectures leads to increased training cost and data requirements, which limit their practicality in data-limited or time-sensitive scenarios.

Model-Space Learning (MSL)

MSL fits each data sample with a designed network, resulting in a fitted model that captures data-inherent dynamics (i.e., changing information). These fitted models then serve as more stable and parsimonious representations of the corresponding data samples. Afterwards, classification or learning methods could be performed on these models instead of the original data. Early studies have used Auto-Regressive Moving Average (Xiong and Yeung 2002) and Hidden Markov Model (Srivastava et al. 2007) to connect neighboring elements in sequences. However, these models, designed for linear systems, struggle to capture nonlinear dynamics. Addressing this, Chen et al. (2013) utilized ESN to fit sequential data. Results demonstrated that when applied to a "next point prediction" task, ESN exhibits considerable performance in capturing non-linear dynamics. Subsequently, ESN-based MSL implementations have found application in diverse areas, including FD of the Barcelona water network (Quevedo et al. 2014), Ground Penetrating Radar (GPR) data (Chen et al. 2023; Zhou et al. 2024), and various types of time-series data (Tang et al. 2025; Chiu and Minku 2022; Liu, Zhou, and Chen 2025).

Conventional MSL relies on step-wise updates with fixed time steps, struggling with irregular/missing data. Recent ODE-integrated approaches improve irregular-sequence fitting (Chen, Zhou, and Chen 2025) but still use fixed networks and non-adjustable metrics, limiting adaptability across scenarios. Additionally, training ODE-integrated networks requires costly gradient backpropagation through numerical solvers, incurring high computational overhead.

Brief Introduction of Echo State Network (ESN)

ESN (Jaeger 2001) is a type of RNN and a typical reservoir computing network, known for its high efficiency in sequential data processing. It comprises an input layer, a fixed recurrent hidden layer (the reservoir), and an output layer. The reservoir, with randomly connected neurons, transforms inputs into high-dimensional "echo states" that capture temporal dependencies and nonlinear dynamics.

To fit a sequence, the input sequence is projected through the input layer and passed into the reservoir step by step, generating a sequence of hidden states. A linear mapping from these states to the output is then learned. During fitting, only the output weights are trained, while the input and reservoir weights are randomly initialized and fixed. The reservoir must satisfy the Echo State Property (ESP)(Chen et al. 2013) to ensure stable and reliable state evolution.

Methodology

Our approach comprises the following three modules:

- **Fitting Sequences via CT-Res.** Each irregular sequence is fitted using CT-Res. The fitted readout model that captures the data-inherent dynamics is then used to represent the original sequence.
- **Adjoint Optimization of CT-Res.** Incorporating and optimizing the "adjoint ESN" that shares aligned structure and parameters as CT-Res, then transferring the optimized parameters and model distance metric to CT-Res for enhanced fitting accuracy and a more "class-discriminative" model space.
- **Fault Diagnosis in Continuous-Time Model Space.** Classifying the fitted models within the optimized model space, thus identifying the corresponding sequence containing faults, along with the fault type.

The contents of these modules are detailed in the following.

Fitting Sequences via CT-Res

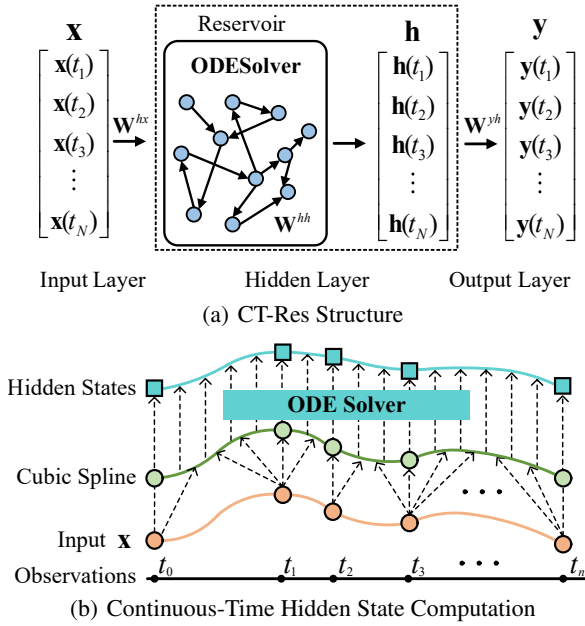
Like typical reservoir computing networks, CT-Res consists of an input layer, a hidden layer, and an output layer, as shown in Fig. 2(a). The hidden layer contains a reservoir of randomly connected neurons, which satisfies the Echo State Property (ESP), allowing the current state to retain information from past inputs while ensuring that older influences decay over time. Unlike conventional discrete-time reservoirs, CT-Res employs continuous-time hidden-state transitions governed by embedding an ODE within the reservoir, which enables it to effectively process irregular sequences.

The processing flow of CT-Res is as follows:

- An input sequence $\mathbf{x}$ is projected through the input layer;
- The projected input is fed into the hidden layer, where the hidden state $\mathbf{h}(t)$ evolves continuously under an ODE, jointly considering the prior state and the current input;
- The resulting hidden states are mapped through the output layer to generate the output values.

The above process allows the hidden state $\mathbf{h}(t)$ to be evaluated at arbitrary time points, supporting a smooth and flexible representation of dynamics under varying time intervals.

In our study, CT-Res uses a leaky integrator (Jaeger et al. 2007) to govern the transition of the hidden state. This mechanism introduces a decay component that allows the reservoir to retain the past information while ensuring that the older information does not dominate. Such balance enables the network's responsiveness to recent inputs while preserv-



(a) CT-Res Structure

(b) Continuous-Time Hidden State Computation

Figure 2: (a) CT-Res comprises input, hidden, and output layers. The hidden layer integrates ODE into the reservoir to produce continuous-time hidden states. (b) An irregular sequence is interpolated via cubic splines to form continuous input. The ODE solver then computes hidden states at arbitrary times based on both the input and preceding states.

ing essential temporal dependencies. The hidden state transitions are described by the following differential equation:

$$\frac{d\mathbf{h}}{dt} = f_\theta(\mathbf{h}, \mathbf{x}) = -a \cdot \mathbf{h} + \tanh(\mathbf{W}^{hh} \cdot \mathbf{h} + \mathbf{W}^{hx} \cdot \mathbf{x}), \quad (1)$$

where $\mathbf{h}$ denotes the hidden state, $\mathbf{W}^{hx}$ is the input weight, $\mathbf{W}^{hh}$ refers to the randomly determined reservoir weights in the hidden layer, which describe neuron connections in the reservoir, and a is a positive scalar that controls the leakage rate. The input and reservoir weights are initially randomly generated, where the reservoir weight should fulfill ESP (Jaeger 2001), with a spectral radius within $(0, 1)$.

To support continuous-time updates, the irregular input sequence $\mathbf{x}(t)$ is first smoothed using cubic spline interpolation (De Boor 1978), allowing input to be estimated at arbitrary time points. Given this smooth input, the ODE is numerically integrated over each interval (t_{n-1}, t_n) using the Runge-Kutta $(4, 5)$ (Dormand and Prince 1980), as shown in Fig. 2 (b). This yields an hidden state $\mathbf{h}(t_n)$:

$$\mathbf{h}(t_n) = \mathbf{ODESolver}(f_\theta(\mathbf{h}, \mathbf{x}), \mathbf{h}(t_{n-1}), (t_{n-1}, t_n)). \quad (2)$$

During the hidden-state update governed by Eqs. (1) and (2), the hidden state evolves under the joint influence of both current input and previous state, modeling temporal dependencies (reflected like "echoes") across time. The continuous integration via the ODE solver allows these dependencies to evolve smoothly over continuous time, maintaining the memory of past states while adaptively responding to new inputs. This continuous-time evolution effectively characterizes the sequence-inherent changing information.

After computing hidden states, the output layer calculates the output value $\mathbf{y}$ using the hidden states:

$$\mathbf{y}(t_n) = \mathbf{W}^{yh} \cdot \mathbf{h}(t_n), \quad (3)$$

where $\mathbf{W}^{yh}$ denotes the output weight.

The fitting process is accomplished via a *next-step prediction* task, aiming to build a mapping from the hidden states to the next observed value. Thus, the output weights $\mathbf{W}^{yh}$ could be solved using ridge regression:

$$\mathbf{W}^{yh} = (\mathbf{H}^\top \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^\top \mathbf{X}, \quad (4)$$

where $\mathbf{H}$ is the horizontally stacked hidden state matrix, and $\mathbf{X}$ contains the target values, i.e., the next-step observations corresponding to each hidden state[2]. The scalar λ is the regularization coefficient, and $\mathbf{I}$ is the identity matrix.

Through the above fitting process, the temporal dependencies within the irregular sequence are effectively captured by CT-Res's continuous-time evolution mechanism. The inherent dynamics of the sequence are encapsulated and represented via a compact fitted readout model:

$$\mathbf{F}(\mathbf{x}) = \mathbf{W}^{yh} \cdot \mathbf{x}, \quad (5)$$

Following this, the readout model serves as a compact representation of the original sequence.

Adjoint Optimization of CT-Res

The varying dynamics across tasks demand adaptive optimization. However, directly optimizing CT-Res via backpropagation introduces substantial computational overhead. CT-Res employs the ODE to describe the continuous-time evolution of hidden states. Direct parameter optimization requires incorporating the ODE solver into gradient backpropagation and backward differentiation through ODE solving. These significantly increase the training time, as evident in our experimental study. Given the limited data and real-time demands of practical FD tasks, such inefficiency makes direct optimization unsuitable.

To efficiently accommodate dynamic variations and form a more "class-discriminative" model space for FD, an adjoint learning strategy is introduced.

- We introduce a discrete-time "adjoint ESN", which shares the same network structure, parameters, and dynamic-capture mechanisms as CT-Res;
- The adjoint ESN is optimized to seek proper input and reservoir weights, ensuring improved fitting accuracy and class discrimination in the model space;
- These optimized shared parameters are then transferred to CT-Res, enabling it to capture continuous-time dynamics tailored to the specific task.

Two objectives are defined for optimizing the shared parameters in CT-Res and the adjoint ESN: 1) Enhancing fitting accuracy on input sequences, and 2) Improving the class

[2]To avoid error propagation, we only use fitting pairs where both $\mathbf{x}(t)$ and $\mathbf{x}(t+1)$ are available. This ensures that the readout is fitted solely on ground-truth values.

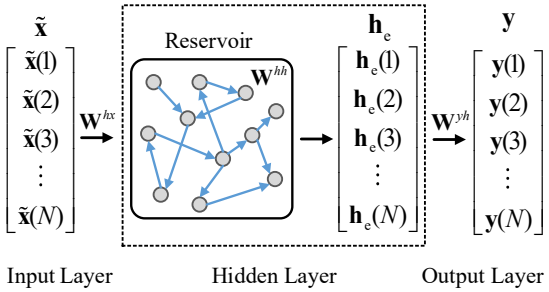


Figure 3: The adjoint ESN shares the same input layer, reservoir, and fitting process as CT-Res. The hidden state iterations are discrete with equal time intervals at each step.

discrimination in the model space. These are formalized as the "*Fitting loss*" and "*Discrimination Loss*", respectively. By optimizing only input and reservoir weights of the adjoint ESN, our method achieves a much smaller parameter scale than existing continuous-time baselines, enabling fast and effective adjoint training under limited training data.

Adjoint Echo State Network (ESN) A discrete-time "adjoint ESN" is introduced, which shares parameters (the input weight and reservoir) with CT-Res, shown as Fig. 3. Concretely, the adjoint ESN takes the cubic spline interpolation $\tilde{\mathbf{x}}(t)$ of the irregular input sequence $\mathbf{x}$ as input. Then, the state evolution in Eq. (1) is approximated by a first-order explicit Euler step over each interval of 1. The hidden state $\mathbf{h}_{\mathrm{e}}(t)$ of adjoint ESN is updated as follows:

$$\begin{aligned}\mathbf{h}_{\mathrm{e}}(t) = \tanh\big(&\mathbf{W}^{hh}\mathbf{h}_{\mathrm{e}}(t-1) + \mathbf{W}^{hx}\tilde{\mathbf{x}}(t)\big) \\ &+ (1-a)\mathbf{h}_{\mathrm{e}}(t-1)\end{aligned}, \tag{6}$$

where the initial state $\mathbf{h}_{\mathrm{e}}(0) = \mathbf{0}$. Similar to Eqs. (3)- (5), a readout model $\mathbf{F}_{\mathrm{e}}$ is fitted via the *next-step prediction* task and applied in the subsequent optimization process.

Incorporating the adjoint ESN enables rapid optimization by "bypassing" the ODE solver. Optimized parameters are then transferred to CT-Res. Despite differences between continuous-/discrete-time hidden state transitions, both CT-Res and the adjoint ESN share identical parameters and dynamic-capture mechanism: they share the same input, reservoir weights, and model-fitting process. Meanwhile, this process does not alter the original data distribution, as interpolated data points are excluded in the next-step prediction task. The adjoint learning strategy leverages the efficiency of discrete-time optimization to pre-estimate parameters under a relatively aligned structure, subsequently transferred to CT-Res for continuous-time dynamic capture, enabling adaptive and efficient parameter optimization.

Distance Metric between Models To quantify the discrepancy between the fitted models derived from different sequences, a model distance metric is first defined. Consider two input sequences $\mathbf{x}_m$ and $\mathbf{x}_n$, which correspond to readout models $\mathbf{F}_m$ and $\mathbf{F}_n$. The distance between these two models is calculated as:

$$\delta(m,n) = \|\mathbf{G}\mathbf{W}_m^{yh} - \mathbf{G}\mathbf{W}_n^{yh}\|_F^2, \tag{7}$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm, computed as the sum of the squared magnitudes of all matrix elements.

In contrast to typical MSL-based approaches that rely on static distance metrics, we incorporate a learnable matrix $\mathbf{G}$ in Eq. (7). It introduces adaptive weights over the elements in $\mathbf{W}^{yh}$, granting the metric flexibility and making it trainable, thus gaining improved ability to differentiate models reflecting distinct underlying dynamics across categories.

Fitting Loss To enhance the fitting ability, we minimize the "Mean Square Error" between the output $\mathbf{y}(t)$ and target $\mathbf{x}(t+1)$. To avoid inaccurate target values caused by interpolation, we fit only at time points where both adjacent values are available in the original irregular sequence. Formally, this results in a fitting loss:

$$L_f = \frac{1}{M}\sum_{m=1}^{M}\sum_{t\in\mathcal{T}_m}\|\mathbf{y}_m(t) - \mathbf{x}_m(t+1)\|_2^2, \tag{8}$$

where subscript m indicates the sample index; $\mathrm{dom}(\mathbf{x}_m)$ denotes the set of all observed time points in $\mathbf{x}_m$; and $\mathcal{T}_m = \{t|t, t+1 \in \mathrm{dom}(\mathbf{x}_m)\}$ contains the time point where the next-step prediction is required.

Discrimination Loss To enhance class-discriminancy in the model space, we propose a linear discrimination loss to minimize the within-class scatter while maximizing the between-class scatter. The weight matrix $\mathbf{W}^{yh}$ of CT-Res readout model is projected by the trainable $\mathbf{G}$ to align with the defined distance: $\mathbf{Z}_m = \mathbf{G}\mathbf{W}_m^{yh}$. Then, the within-class scatter $\mathbf{S}_w$ and between-class scatter $\mathbf{S}_b$ are computed as:

$$\begin{aligned}\mathbf{S}_w &= \sum_{c=1}^{C}\sum_{m:l_m=c}\big(\mathbf{Z}_m - \boldsymbol{\mu}_c\big)\big(\mathbf{Z}_m - \boldsymbol{\mu}_c\big)^{\mathsf{T}}, \\ \mathbf{S}_b &= \sum_{c=1}^{C} m_c\big(\boldsymbol{\mu}_c - \boldsymbol{\mu}\big)\big(\boldsymbol{\mu}_c - \boldsymbol{\mu}\big)^{\mathsf{T}},\end{aligned} \tag{9}$$

where l_m denotes the label of m-th sample; m_c is the number of samples in class c; $\boldsymbol{\mu}_c$ is the mean of $\mathbf{Z}_m$ belonging to class c; and $\boldsymbol{\mu}$ is the overall mean. Consequently, our objective is to minimize the following loss:

$$L_d = \frac{\mathrm{Tr}(\mathbf{S}_w)}{\mathrm{Tr}(\mathbf{S}_b) + \epsilon}. \tag{10}$$

Here, $\mathrm{Tr}()$ is trace operator and ϵ ensures numerical stability.

Overall Training Process Combining Eqs. (8) and (10), we construct the overall loss function:

$$L_{\text{total}} = L_f + \lambda L_d, \tag{11}$$

where $\lambda > 0$ controls the trade-off between L_f and L_d. We minimize L_{total} via gradient descent, optimizing the shared parameters (i.e., $\mathbf{W}^{hx}$ and $\mathbf{W}^{hh}$) of the adjoint ESN and the weight matrix $\mathbf{G}$ for the distance metric. The learned parameters and distance metric are then transferred to CT-Res, enhancing its fitting accuracy and class discriminability in the resulting model space. Moreover, since only $\mathbf{W}^{hx}$ and $\mathbf{W}^{hh}$ are trained, the optimization remains lightweight and efficient, generalizing well even with limited training data.

Fault Diagnosis in Continuous-Time Model Space

FD involves identifying if a newly collected sequence is normal or associated with a specific fault type. Our approach involves the "*Training Phase*" and "*Diagnosis Phase*".

Training Phase CT-Res and its adjoint ESN are built with identical structure, parameters, and dynamic-capture mechanism. Using training sequences with labels, the adjoint optimization strategy efficiently updates shared parameters and the model distance metric. These optimized components are transferred to CT-Res, which subsequently fits training sequences to derive corresponding readout models, mapping the original data into the continuous-time model space. A classifier (e.g., KNN/SVM) is trained on these models supported with the model distance metric.

Diagnosis Phase Newly collected sequences are fitted using the above-optimized CT-Res to derive their readout models, thus mapped into the model space. These models are then classified using the pre-trained classifier. Based on the classification result, each input sequence is identified as either normal or indicative of a specific fault type.

Experimental Study

This section presents experiments under irregular sampling and limited data, including comparison and analysis.

Experimental Settings

Experiments are conducted on an Intel(R) Core(TM) i7-14700kF CPU and an NVIDIA GeForce RTX 4090 GPU. The reservoir size is 50 with an initial spectral radius of 0.1; leaky rate is 1.0; regularization λ is 1. Adjoint training uses AdamW with learning rate 0.001, weight decay 0.001, and batch size 64. The model-space classifier adopts the official SVM implementation from https://scikit-learn.org/.

Dataset Settings Four datasets are utilized: **CWRU** (subsets A–D and combined E, each with 200 samples per class, 2048 time steps, 10 classes: 1 normal, 9 faults); **SU** (subsets G20/G30 for gearboxes and B20/B30 for bearings, 1000 samples per class, 1024 time steps, 5 classes: 1 normal, 4 faults); **WHU** (45 samples per class, 2048 time steps, 4 classes: 1 normal, 3 faults); and **TBV** (100 samples per class, 1024 time steps, 13 classes: 1 normal, 12 faults).

By default, we set **1) Irregular Sampling:** 50% of data points are randomly removed per sequence; **2) Limited Training Data:** The training set includes 200 sequences per sub-dataset for CWRU and SU, 90 for WHU, and 130 for TBV; the test set includes 1800, 4800, 90, and 1300 sequences for CWRU, SU, WHU, and TBV, respectively.

Baseline Methods The baselines include: **1) Traditional machine learning**: **NFFT-SVM** (Keiner, Kunis, and Potts 2009), which applies a non-uniform FFT for feature extraction followed by SVM; **2) RNN-based methods**: **GRU-Δt**, **GRU-Impute**, and **GRU-D** (Che et al. 2018), address irregular sampling by incorporating time intervals, applying uniform-grid interpolation, or introducing exponential decay mechanisms, respectively. **3) Transformer adaptations**: **mTANs** (Shukla and Marlin 2021) and **Warpformer** (Zhang et al. 2023b), leverage attention strategies to handle irregular sequences; **4) Continuous-time approaches**: **ODE-RNN**, **Latent ODE** (Rubanova, Chen, and Duvenaud 2019), and **Neural CDE** (Kidger et al. 2020), integrate Neural ODEs with RNNs or VAEs to construct continuous latent states. **5) Ablation Implementation**: **Adjoint ESN**, uses the optimized adjoint ESN to fit the interpolated data and classify the fitted readout model; **CT-Res (w/o)**, represents directly classifying the fitted CT-Res models without adjoint optimization (w/o).

Experimental Results and Discussions

Table 1 reports that traditional NFFT-SVM fails to distinguish the dynamics of irregular sequences, resulting in poor classification. RNN-based methods and attention-based adaptations also struggle under sparse, non-uniform inputs and limited data. Continuous-time baselines and Warpformer, while modeling continuous-time dynamics and flexible alignment, still fall short. In contrast, our approach consistently achieves superior results: CT-Res integrates ODE into the hidden layer to capture continuous-time dynamics without relying on fixed time steps, and the adjoint optimization incorporates a discrete-time adjoint ESN for efficient parameter and metric optimization. As Fig. 4 illustrates, these enable class-wise clustering in the model space, supporting reliable FD even with limited training data.

Different Missing Rates To assess robustness, we evaluate the top-five methods (from Table 1) under varying missing rates[3] (10% to 70%). As shown in Table 2, continuous-time baselines like ODE-RNN and Neural CDE struggle to capture data-inherent dynamics under limited training data and severe sparsity. In contrast, our method maintains robustness via an ODE-embedded reservoir for continuous-time dynamic capture, enhanced by adjoint learning for efficient optimization. Even in the extreme case of 70% missing rates, it achieves over 80% accuracy on most datasets.

Training Time Table 3 reports the training times[4] of the top-five methods. Continuous-time baselines like Neural CDE and ODE-RNN require backpropagation through ODE solvers, leading to significantly longer training, especially for long sequences (e.g., CWRU with 2048 time steps incurs the highest cost). Our approach leverages a discrete-time adjoint ESN for efficient optimization without ODE solvers. With far fewer trainable parameters (only input and reservoir weights), it achieves lightweight and efficient training.

Ablation Study Our approach features: (1) Integrating ODE into the hidden layer for continuous-time dynamic capture, and (2) Adjoint optimization with a discrete-time adjoint ESN for efficient training, bypassing ODE solvers.

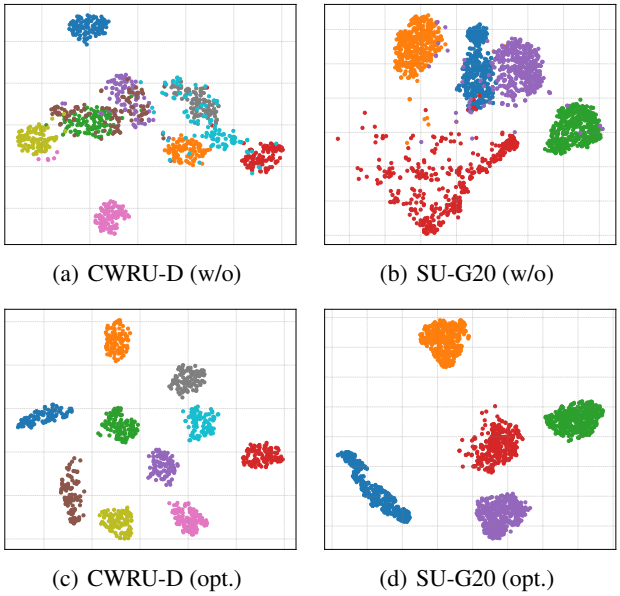
For the first, we examine a variant: *Adjoint ESN*, which replaces CT-Res with its discrete-time counterpart (i.e., the

[3]Missing rate refers to the proportion of removed points.

[4]Training time refers to the duration from feeding in training data to enabling FD on new sequences. Inference for each new sequence remains within 0.1s, satisfying practical requirements.

Methods	CWRU					SU				WHU	TBV
	A	B	C	D	E	G20	G30	B20	B30		
NFFT-SVM	31.64	25.24	23.31	24.07	19.92	82.67	70.37	78.89	65.56	35.59	33.91
GRU-Δt	77.11	39.28	58.89	60.06	65.74	93.39	89.88	95.56	96.23	66.31	21.47
GRU-Impute	14.61	15.17	14.78	18.22	24.92	89.78	82.86	89.70	88.65	78.47	28.86
GRU-D	24.17	30.39	29.83	27.33	57.58	43.67	30.00	50.33	49.22	32.22	21.51
Latent-ODE	81.17	72.78	70.17	72.28	83.35	83.58	89.85	74.58	75.88	90.37	56.40
ODE-RNN	82.22	84.17	80.28	85.33	91.75	62.12	78.60	82.16	91.21	89.78	53.43
Neural CDE	78.39	82.83	86.17	84.61	76.66	77.98	69.41	92.43	93.02	77.77	57.46
mTANs	11.98	12.65	12.77	13.06	11.34	84.56	64.73	75.25	61.11	36.23	23.56
Warpformer	79.33	79.77	70.00	87.61	71.46	92.33	59.92	97.06	99.14	80.00	60.36
Adjoint ESN	71.20	73.78	75.56	78.08	71.88	91.32	93.38	97.68	96.80	90.94	62.29
CT-Res (w/o)	81.56	83.44	85.50	87.05	85.76	92.33	91.67	94.37	95.22	92.23	64.59
Proposed Approach	**88.72**	**92.44**	**94.50**	**96.61**	**92.09**	**96.22**	**95.64**	**98.97**	**99.25**	**96.56**	**78.76**

Table 1: Classification Accuracy (%) of Proposed Approach and Baselines on Adopted Datasets



(a) CWRU-D (w/o) (b) SU-G20 (w/o)

(c) CWRU-D (opt.) (d) SU-G20 (opt.)

Figure 4: t-SNE visualizations of the readout models fitted on partial samples from the CWRU-D and SU-G20 using CT-Res with adjoint optimization (opt.) and without optimization (w/o). Blue indicates normal, others represent fault types. Adjoint optimization significantly enhances class-discriminancy within the model space.

	Missing Rate	ODE RNN	Latent ODE	Neural CDE	Warp former	Proposed Approach
CWRU Avg	10%	88.46	86.43	87.04	79.87	96.92
	30%	87.53	78.64	83.52	78.98	94.17
	50%	82.74	75.95	81.73	77.63	92.87
	70%	78.16	71.86	72.54	75.59	81.34
SU Avg	10%	86.50	86.80	91.98	91.94	99.09
	30%	84.50	84.61	85.20	89.48	98.73
	50%	78.66	80.97	83.21	87.11	97.52
	70%	57.55	77.78	71.22	84.42	89.34
WHU	10%	83.33	78.89	86.66	81.11	98.89
	30%	79.67	71.11	85.55	80.00	97.77
	50%	75.00	67.78	77.77	80.00	96.56
	70%	71.11	65.56	61.11	77.77	90.00
TBV	10%	72.62	73.10	84.60	83.28	95.58
	30%	65.24	62.38	71.19	77.00	88.17
	50%	51.67	54.76	57.46	60.36	78.76
	70%	35.60	32.14	36.58	45.41	51.29

Table 2: Accuracy (%) of Better-performed Methods at Various Missing Rates on Different Datasets

Dataset	ODE RNN	Latent ODE	Neural CDE	Wrap former	CT-Res (b/p)	Proposed Approach
CWRU	6841.93	6969.19	3380.91	149.11	1954.37	10.24
SU	3565.62	3860.24	697.63	63.79	1056.51	7.95
WHU	1067.11	1406.98	1679.23	41.75	686.28	3.72
TBV	1833.19	2355.12	1657.17	78.56	879.34	4.84

Table 3: Average Training Time(s) of Better-performed Methods on Different Datasets (50% Missing Rate)

adjoint ESN) for data fitting and representation, while retaining the same optimization process and model-space classifier. As shown in Table 1, Adjoint ESN yields inferior results, indicating that discrete approximations are insufficient to capture continuous-time dynamics of irregular sequences.

For the second, we consider two variants: *CT-Res (w/o)*, which removes adjoint optimization, and *CT-Res (b/p)*, which applies direct backpropagation to optimize CT-Res. Table 1 indicates that CT-Res (w/o) achieves moderate accuracy, confirming the effectiveness of ODE-embedded dynamic capture but lacking sufficient class separation (Fig. 4). As Table 3 shows, CT-Res (b/p) requires ODE solver calls at each training iteration, resulting in slow training comparable to other continuous-time baselines. In contrast, our full approach combines strong performance and efficiency, validating the complementary benefits of both attributes.

Conclusion

This study proposes FD of irregular sequences by learning in the continuous-time model space, introducing CT-Res for irregular-sequence fitting, along with an adjoint optimization strategy. CT-Res captures continuous-time dynamics via an ODE-embedded hidden layer, while adjoint optimization enables efficient training without ODE solvers. Experiments across multiple FD benchmarks demonstrate its accuracy, efficiency, and robustness under irregular and limited data.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62576327, 62206261, 62137002, 62176245), in part by the Fundamental Research Funds for the Central Universities (No. WK2150110039).

References

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1): 6085.

Chen, A.; Zhou, X.; and Chen, H. 2025. Efficient Anomaly Detection of Irregular Sequences in Ct-Echo Model Space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 15731–15739.

Chen, A.; Zhou, X.; Fan, Y.; and Chen, H. 2023. Underground diagnosis based on gpr and learning in the model space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 3832–3844.

Chen, H.; Tiňo, P.; Rodan, A.; and Yao, X. 2013. Learning in the model space for cognitive fault diagnosis. *IEEE transactions on neural networks and learning systems*, 25(1): 124–136.

Chen, H.; Tiňo, P.; and Yao, X. 2014. Cognitive fault diagnosis in Tennessee Eastman Process using learning in the model space. *Computers & chemical engineering*, 67: 33–42.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chiu, C. W.; and Minku, L. L. 2022. A Diversity Framework for Dealing With Multiple Types of Concept Drift Based on Clustering in the Model Space. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3): 1299–1309.

Chowdhury, R. R.; Li, J.; Zhang, X.; Hong, D.; Gupta, R. K.; and Shang, J. 2023. PrimeNet: Pre-Training for Irregular Multivariate Time Series. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

De Boor, C. 1978. *A practical guide to splines*, volume 27. Springer-Verlag New York.

Dormand, J. R.; and Prince, P. J. 1980. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1): 19–26.

Jaeger, H. 2001. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34): 13.

Jaeger, H.; Lukoševičius, M.; Popovici, D.; and Siewert, U. 2007. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3): 335–352.

Jhin, S. Y.; Lee, J.; and Park, N. 2023. Precursor-of-Anomaly Detection for Irregular Time Series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 917–929.

Keiner, J.; Kunis, S.; and Potts, D. 2009. Using NFFT 3—A Software Library for Various Nonequispaced Fast Fourier Transforms. *ACM Trans. Math. Softw.*, 36(4).

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707.

Liu, S.; Zhou, X.; and Chen, H. 2025. Multiscale temporal dynamic learning for time series classification. *IEEE Transactions on Knowledge and Data Engineering*.

Ma, Q.; Li, S.; Zhuang, W.; Wang, J.; and Zeng, D. 2020. Self-supervised time series clustering with model-based dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9): 3942–3955.

Quevedo, J.; Chen, H.; Cugueró, M. À.; Tino, P.; Puig, V.; Garciá, D.; Sarrate, R.; and Yao, X. 2014. Combining learning in model space fault diagnosis with data validation/reconstruction: Application to the Barcelona water network. *Engineering Applications of Artificial Intelligence*, 30: 18–29.

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

Shukla, S. N.; and Marlin, B. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *International Conference on Learning Representations*.

Srivastava, P. K.; Desai, D. K.; Nandi, S.; and Lynn, A. M. 2007. HMM-ModE–Improved classification using profile hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences. *BMC bioinformatics*, 8(1): 1–17.

Sterne, J. A.; White, I. R.; Carlin, J. B.; Spratt, M.; Royston, P.; Kenward, M. G.; Wood, A. M.; and Carpenter, J. R. 2009. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338.

Tang, Z.; Zhou, X.; Liu, S.; Wei, C.; Chen, A.; and Chen, H. 2025. Learning in the Model Space: Fault Diagnosis by Co-objective Learning in DynInt Model Space. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Xiong, Y.; and Yeung, D.-Y. 2002. Mixtures of ARMA models for model-based time series clustering. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, 717–720. IEEE.

Yuan, Z.; Ban, X.; Zhang, Z.; Li, X.; and Dai, H.-N. 2023. ODE-RSSM: Learning Stochastic Recurrent State Space Model from Irregularly Sampled Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 11060–11068.

Zhang, J.; Zheng, S.; Cao, W.; Bian, J.; and Li, J. 2023a. Warpformer: A Multi-Scale Modeling Approach for Irregular Clinical Time Series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3273–3285. Association for Computing Machinery.

Zhang, J.; Zheng, S.; Cao, W.; Bian, J.; and Li, J. 2023b. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3273–3285.

Zhou, X.; Liu, S.; Chen, A.; and Chen, H. 2024. Learning in CubeRes model space for anomaly detection in 3D GPR data. In *Proc. 33rd Int. Joint Conf. Artif. Intell*, 5662–5670.