FlowPath: Learning Data-Driven Manifolds with Invertible Flows for Robust Irregularly-sampled Time Series Classification

YongKyung Oh[1], Dong-Young Lim[2,3*], Sungil Kim[2,3*]

[1]Medical & Imaging Informatics (MII) Group, University of California, Los Angeles (UCLA), CA, USA
[2]Department of Industrial Engineering, Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
[3]Artificial Intelligence Graduate School, Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
yongkyungoh@mednet.ucla.edu, {dlim, sungil.kim}@unist.ac.kr

Abstract

Modeling continuous-time dynamics from sparse and irregularly-sampled time series remains a fundamental challenge. Neural controlled differential equations provide a principled framework for such tasks, yet their performance is highly sensitive to the choice of control path constructed from discrete observations. Existing methods commonly employ fixed interpolation schemes, which impose simplistic geometric assumptions that often misrepresent the underlying data manifold, particularly under high missingness. We propose *FlowPath*, a novel approach that learns the geometry of the control path via an invertible neural flow. Rather than merely connecting observations, FlowPath constructs a continuous and data-adaptive manifold, guided by invertibility constraints that enforce information-preserving and well-behaved transformations. This inductive bias distinguishes FlowPath from prior unconstrained learnable path models. Empirical evaluations on 18 benchmark datasets and a real-world case study demonstrate that FlowPath consistently achieves statistically significant improvements in classification accuracy over baselines using fixed interpolants or non-invertible architectures. These results highlight the importance of modeling not only the dynamics along the path but also the geometry of the path itself, offering a robust and generalizable solution for learning from irregular time series.

Code — https://github.com/yongkyung-oh/FlowPath
Extended version — https://arxiv.org/abs/2511.10841

Introduction

Many real-world systems evolve continuously over time. In contrast, observations of these systems are typically sparse, irregular, and incomplete (Choi et al. 2016; Hossain, Ahad, and Inoue 2020; Oh, Lim, and Kim 2024). Such data are referred to as irregularly-sampled time series (ISTS), characterized by non-uniform sampling intervals and missing observations over time. While discrete-time models such as GRUs (Chung et al. 2014) have been adapted to handle irregularity using decay mechanisms (Che et al. 2018), they remain fundamentally discrete approximations. Neural Differential Equations (NDEs) address this limitation by modeling time as a continuous variable, with Neural Ordinary Differential Equations (Neural ODEs) as a foundational case (Chen et al. 2018; Lu et al. 2018).

For ISTS, Neural Controlled Differential Equations (Neural CDEs) extend this idea by modeling system dynamics as driven by a continuous control path derived from the input (Kidger et al. 2020). Unlike Neural ODEs that evolve from an initial state (Rubanova, Chen, and Duvenaud 2019), Neural CDEs respond directly to the full input stream. However, the control path is typically constructed via fixed interpolation methods, which impose predefined geometric assumptions. This choice has been shown to affect performance substantially and remains a key limitation of the Neural CDE framework (Morrill et al. 2022).

In this paper, we propose *FlowPath*, a framework that models the control path using an invertible neural flow (Dinh, Sohl-Dickstein, and Bengio 2017; Papamakarios et al. 2021). This replaces the fixed interpolant with a learned, data-adaptive transformation that preserves invertibility. Such flows serve as universal approximators of diffeomorphisms (Teshima et al. 2020; Ishikawa et al. 2023), allowing the learned path to be smooth and information-preserving. We formalize the FlowPath architecture, analyze its theoretical properties, and evaluate it on 18 benchmark datasets and real-world datasets. Our results demonstrate that learning a structurally-constrained, data-driven manifold leads to statistically significant improvements in classification accuracy over baselines using either fixed interpolants or non-invertible learned paths.

Related Work

RNN-style Discretizations. Traditional Recurrent Neural Networks (RNNs) assume fixed step sizes (Goodfellow, Bengio, and Courville 2016), and adaptations such as GRU-D (Che et al. 2018) incorporate missingness indicators or time-lag decay to handle irregular sampling, but remain discrete-time approximations that can still struggle with large sampling gaps or nonuniform intervals. Thus, a conventional RNN or GRU-D processes data at sampled time points $\{t_k\}$. An RNN state $\boldsymbol{h}_k \in \mathbb{R}^{d_h}$ typically evolves via:

$$\boldsymbol{h}_{k+1} = \Psi\Big(\boldsymbol{h}_k,\ \boldsymbol{x}_{k+1},\ \Delta t_k\Big),$$

where $\boldsymbol{x}_{k+1}$ is the observation at time t_{k+1} and $\Delta t_k = t_{k+1} - t_k$. The function Ψ may incorporate gating or de-

*Corresponding Authors

cay terms to handle missingness and so remains a discrete approximation of an underlying continuous process.

NDE-based Models. To overcome the limitations of purely discrete-time methods, Chen et al. (2018) introduced Neural ODEs, casting hidden state evolution as a continuous function of time,

$$\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = f\big(t, \boldsymbol{z}(t)\big), \quad \boldsymbol{z}(0) = h\big(\boldsymbol{x}(0)\big).$$

The state $\boldsymbol{z}(t) \in \mathbb{R}^{d_z}$ in a Neural ODE is obtained by integrating an initial value $\boldsymbol{z}(0)$ from 0 to T. While this continuous viewpoint elegantly handles irregular time points, its strict reliance on the initial value problem makes it difficult to incorporate new observations arriving at later times without restarting or extending the integration (Rubanova, Chen, and Duvenaud 2019; Kidger 2022).

Neural CDEs (Kidger et al. 2020) overcome this limitation by representing the input time series as a control path $X(t) \in \mathbb{R}^{d_x}$, thereby enabling the hidden state to evolve continuously in response to incoming observations. The hidden state dynamics are governed by a Riemann–Stieltjes integral with respect to the control path:

$$\boldsymbol{z}(t) = \boldsymbol{z}(0) + \int_0^t f\big(\tau, \boldsymbol{z}(\tau)\big)\,\mathrm{d}X(\tau), \quad \boldsymbol{z}(0) = h\big(X(0)\big).$$

Control path $X(t)$ is typically constructed from observed data using fixed interpolation methods such as linear or cubic splines (Morrill et al. 2021). This introduces a structural prior that may not align with the true dynamics. Empirical results show that interpolation choice is a sensitive hyperparameter with significant impact on performance, and no single method is optimal across tasks (Morrill et al. 2022).

Recent studies replace the fixed interpolant with a learned path from unconstrained networks, such as encoder-decoders (Jhin et al. 2022, 2023). Others apply attention mechanisms within the Neural CDE framework to reweight temporal contributions (Jhin et al. 2024). Further work focuses on improving the expressivity of the dynamics by signature transformation (Morrill et al. 2021). These approaches enhance flexibility but may introduce instability when the learned path lacks structural constraints.

Neural Flows. Bilos et al. (2021) propose modeling ODE solutions with an invertible transformation, circumventing numerical solvers by directly parameterizing a diffeomorphism in time. Such invertible mappings have been extensively studied in normalizing flows (Papamakarios et al. 2021), where tractable Jacobians enable density estimation or generative sampling. Formally, Neural Flow (Bilos et al. 2021) is designed to learn an invertible mapping $F(t, \boldsymbol{x}(0); \theta_F)$ that directly parameterizes the solution of an ODE or dynamical process. Given an implicit dynamical system governed by,

$$\frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} = f\big(\boldsymbol{z}(t)\big), \quad \boldsymbol{z}(0) = \boldsymbol{x}(0),$$

Neural Flow posits a neural function F satisfying,

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\, F(t, \boldsymbol{x}(0)) = f\big(F(t, \boldsymbol{x}(0))\big), \quad F(0, \boldsymbol{x}(0)) = \boldsymbol{x}(0).$$

Crucially, $F(\cdot)$ is designed to be a diffeomorphism in $\boldsymbol{x}$, ensuring that mapping local volumes is always invertible. This property allows Neural Flow to avoid the need for standard numerical ODE solvers and to guarantee the model's Jacobian is tractable. While originally aimed at continuous normalizing flows for generative tasks, the concept of applying invertible flows to dynamic time series modeling has grown (Bilos et al. 2021; Oh, Lim, and Kim 2025).

Our approach builds on the observation that Neural CDEs and Neural Flows address complementary aspects of time series modeling. Instead of using a Neural Flow as an end-to-end model, we use it to construct a data-driven control path. This design combines the continuous-time formulation of NDEs with a learnable, invertible transformation that adapts to the observed data, avoiding the limitations of fixed interpolation schemes or unconstrained trajectories.

Methodology

We are given a dataset $\mathcal{D} = \{(\boldsymbol{X}_i, y_i)\}_{i=1}^{N}$, where each instance $\boldsymbol{X}_i$ comprises a time-indexed sequence $\big(t_i, \boldsymbol{x}_i\big)$. Each $t_i \subset \mathbb{T}$ represents the set of sampling times, and $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ is the multivariate observation at those times. We aim to classify each instance with a label $y_i \in \{1, 2, \ldots, K\}$. The sampling intervals $\Delta t_j = t_{j+1} - t_j$ are not necessarily uniform, and certain intervals may be missing data entirely. We aim to learn a predictive function $\mathcal{H} : (t_i, \boldsymbol{x}_i) \mapsto y_i$ that robustly classifies sequences under irregular sampling.

FlowPath Framework

Raw observation $\boldsymbol{x}(t)$ includes how missing values create discontinuities and distortions in the time series, disrupting its temporal structure. The proposed learnable flow $\Phi(t)$ generates a continuous path that preserves the underlying temporal dynamics. Unlike interpolation, which applies a fixed approach, the learnable flow adapts to surrounding patterns, leading to more informative representations for the downstream task.

Learning the Underlying Manifold. Let $F\colon [0, T] \times \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ be a neural network parameterized by θ_F such that the mapping F is a diffeomorphism in its second argument (Bilos et al. 2021). We explicitly define the learnable control path $\Phi : [0, T] \to \mathbb{R}^{d_x}$ as:

$$\Phi(t) = F\big(t,\, \boldsymbol{x};\, \theta_F\big), \tag{1}$$

where $\boldsymbol{x}(0) \in \mathbb{R}^{d_x}$ denotes the earliest available observation in the irregular time series. Unlike fixed spline-based interpolations, $\Phi(t)$ is parameterized by a neural network F with learnable parameters θ_F, allowing the control path to be learned directly from data in a flexible and adaptive manner.

Dynamics on the Learned Manifold. We define a hidden state $\boldsymbol{z}(t) \in \mathbb{R}^{d_z}$ governed by the Riemann–Stieltjes integral as suggested by Kidger et al. (2020):

$$\boldsymbol{z}(t) = \boldsymbol{z}(0) + \int_0^t f\big(\tau, \boldsymbol{z}(\tau); \theta_f\big)\,\mathrm{d}\Phi(\tau). \tag{2}$$

Then, an equivalent ODE form of Eq. (2) defines:

$$\dot{\boldsymbol{z}}(t) = f\big(t, \boldsymbol{z}(t); \theta_f\big)\,\dot{\Phi}(t), \tag{3}$$

with $\boldsymbol{z}(0) = h(\boldsymbol{x}(0))$, and $h\colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ an embedding layer. In Eq. (3), the dot notation $\dot{\boldsymbol{z}}(t)$ and $\dot{\Phi}(t)$ denote the time derivatives, i.e., $\frac{d\boldsymbol{z}(t)}{dt}$ and $\frac{d\Phi(t)}{dt}$, respectively. The smoothness assumptions on Φ guarantee Lipschitz continuity in time, while invertibility enforces distribution-preserving transformations (Papamakarios et al. 2021; Bilos et al. 2021). Since Φ is a diffeomorphism, the learned path $t \mapsto \Phi(t)$ admits well-defined inverses and Jacobians, which strengthen theoretical guarantees such as existence, uniqueness, and universal approximation for continuous-time processes. In practice, numerical solvers treat $\dot{\Phi}(t)$ as a known function (output of a neural net) and integrate $\dot{\boldsymbol{z}}(t)$ forward in time, following the ODE form in Eq. (3).

Considerations for the Invertible Flow. We parameterize the control path using an invertible neural network based on normalizing flows, which provide a flexible framework for learning smooth, bijective transformations (Papamakarios et al. 2021). Recent theoretical results show that such architectures can approximate any diffeomorphism under suitable conditions (Teshima et al. 2020; Ishikawa et al. 2023). This ensures that the learned path $\Phi(t)$ remains invertible and smooth by design, reducing the risk of instability often seen in unconstrained path parameterizations.

Within the FlowPath framework, $\Phi(t) = F(t, \boldsymbol{x}(0); \theta_F)$ acts as a trainable control path that modulates the evolution of the latent state $\boldsymbol{z}(t)$. The function F is designed as an invertible neural network, drawing inspiration from normalizing flows (Papamakarios et al. 2021) and specifically neural flows that parameterize diffeomorphisms (Bilos et al. 2021). This design ensures that $\Phi(t)$ is a smooth transformation of its inputs, providing FlowPath with enhanced flexibility to model complex temporal dependencies.

Properties of FlowPath

The design of FlowPath results in a time-continuous model that (i) preserves information through an invertible flow, (ii) ensures well-posedness of the underlying dynamics, and (iii) promotes strong generalization capability. We present these three properties below, detailing full explanations in the supplementary material.

FlowPath uses an invertible flow Φ that reshapes the latent distribution without collapsing or tearing probability mass. The following theorem states that the instantaneous change in log-density is exactly driven by the divergence of the controlled dynamics, ensuring no unintended compression or expansion of probability mass.

Theorem 1 (Preservation of Probability Density)**.** *Under the FlowPath framework in Eq. (2), let $\Phi(t)$ be a C^1-diffeomorphism, and f be Lipschitz continuous in $\boldsymbol{z}$. Then the probability density $p(\boldsymbol{z}(t))$ of the latent state evolves according to*

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p(\boldsymbol{z}(t)) \;=\; -\operatorname{div}_{\boldsymbol{z}}\Big(f\big(t, \boldsymbol{z}(t)\big)\, \dot{\Phi}(t)\Big). \tag{4}$$

Consequently, Φ neither collapses nor arbitrarily expands probability mass, preserving the geometry of the latent distribution.

This property, while crucial for continuous normalizing flows in generative modeling (Papamakarios et al. 2021), also benefits our classification task by preserving class-separating structures in latent space, thereby improving robustness to irregular sampling and missing data.

Next, Theorem 2 establishes the standard existence and uniqueness guarantee for the FlowPath dynamics.

Theorem 2 (Existence and Uniqueness of Solutions)**.** *Fix θ_f. Let $f(t, \cdot; \theta_f)$ in Eq. (2) be continuous in t and Lipschitz in $\boldsymbol{z}$, and $\Phi(t)$ be continuously differentiable on $[0, T]$. Then for any initial condition $\boldsymbol{z}(0) \in \mathbb{R}^{d_z}$, there exists a unique continuous solution*

$$\boldsymbol{z}(t) = \boldsymbol{z}(0) + \int_0^t f\big(s, \boldsymbol{z}(s); \theta_f\big)\, \mathrm{d}\Phi(s).$$

The universal approximation properties of Neural ODEs and Neural Flows have been extensively studied and established in prior work (Lin and Jegelka 2018; Li, Lin, and Shen 2022; Teshima et al. 2020; Ishikawa et al. 2023). Building upon these results, one can check that FlowPath is a universal approximator under appropriate regularity conditions. Under the assumption that Φ is a C^1-diffeomorphism and f is Lipschitz, Neural ODEs are universal approximators for any smooth target trajectory u in the uniform norm. Thus, one can choose f such that within any desired precision ϵ'

$$\sup_{t \in [0,T]} \big\| f(t, z(t)) - \dot{\Phi}(t)^{-1} \dot{u}(t) \big\| < \epsilon'.$$

Substituting this into the FlowPath dynamics $\dot{z}(t) = f(t, z(t))\, \dot{\Phi}(t)$ and integrating yields

$$\begin{aligned}
\sup_{t \in [0,T]} \|z(t) - u(t)\| &\le \int_0^T \|f(\tau, z(\tau))\, \dot{\Phi}(\tau) - \dot{u}(\tau)\|\, \mathrm{d}\tau \\
&\le \int_0^T \Big(\sup_{s \in [0,T]} \|\dot{\Phi}(s)\|\Big)\, \epsilon'\, \mathrm{d}\tau \\
&= \epsilon' \Big(\sup_{s \in [0,T]} \|\dot{\Phi}(s)\|\Big)\, T.
\end{aligned}$$

Since $\sup_{t \in [0,T]} \|\dot{\Phi}(t)\|$ and T are finite, by choosing ϵ' appropriately, FlowPath can approximate any smooth trajectory to within arbitrary tolerance $\epsilon > 0$.

Let $\mathcal{L}$ be an ℓ-Lipschitz loss in the model's final output. We parameterize FlowPath by $\theta = (\theta_f, \theta_F)$, where θ_f specifies the vector field f and θ_F defines the invertible flow Φ.

Theorem 3 (Generalization Bound)**.** *Assume that the flow map $\Phi(\cdot; \theta_F)\colon [0, T] \to \mathbb{R}^{d_x}$ is a diffeomorphism with $\sup_{t \in [0,T]} \|\dot{\Phi}(t)\| \le M_\Phi$, and the vector field $f(\cdot, \cdot; \theta_f)$ is L_f-Lipschitz in the hidden state. Let $\mathcal{F}_\Theta$ be the class of all such FlowPath predictors. Given n i.i.d. samples $\{(X_i, y_i)\}_{i=1}^n$ drawn from $\mathcal{P}$, define the population risk by*

$$\mathcal{R}_{\text{true}}(f) = \mathbb{E}_{(X,y)\sim\mathcal{P}}\big[\mathcal{L}\big(f(X), y\big)\big],$$

and the empirical risk by

$$\mathcal{R}_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}\big(f(X_i), y_i\big).$$

If $\widehat{f} = \arg\min_{f \in \mathcal{F}_\Theta} \mathcal{R}_{\text{emp}}(f)$*, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\mathcal{R}_{\text{true}}(\widehat{f}) - \mathcal{R}_{\text{emp}}(\widehat{f})\right| \leq \alpha \frac{1}{\sqrt{n}} + \sqrt{\frac{\ln(1/\delta)}{2n}},$$

where α *depends on* ℓ*,* L_f*, and* M_Φ*.*

This result, derived via Rademacher complexity arguments (Bartlett, Foster, and Telgarsky 2017; Chen, Li, and Zhao 2020), indicates that, as the number of training samples n grows, the empirical loss of a FlowPath model closely approximates its true expected loss on unseen data.

Experiments

We design experiments to evaluate both the classification accuracy and robustness of our model under varying degrees of missing observations. Specifically, we consider 18 diverse time series classification datasets from UEA & UCR Data Repository, referenced in (Bagnall et al. 2018; H. A. Dau et al. 2019), utilizing the `sktime` Python library (Löning et al. 2019). We followed the experimental setup and dataset selection described by Oh, Lim, and Kim (2025). The original paper considered three categories: 'Motion & HAR', 'Electrocardiogram (ECG) & Electroencephalogram (EEG)', and 'Sensor' domains, as summarized in the supplementary material.

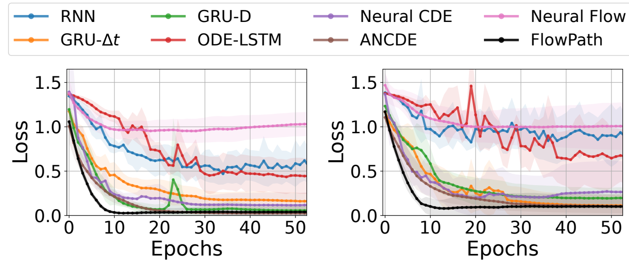
In the experiments, three derivative datasets were created from each original dataset by artificially inducing missing data at rates of 30%, 50%, and 70%. Consequently, for each original dataset, there were four distinct settings, including the original data. The datasets were split into training, validation, and testing subsets, adhering to a 70/15/15 proportion. Lastly, the classification metrics were computed for each setting and averaged with five iterations.

Benchmark Methods

We utilized the conventional *RNN* (Medsker and Jain 1999), *LSTM* (S. Hochreiter and J. Schmidhuber 1997), and *GRU* (Chung et al. 2014), with mean imputation. Furthermore, we employed several modifications of the GRU, specifically *GRU-*Δt (Choi et al. 2016), *GRU-D* (Che et al. 2018), which have been tailored to efficiently handle ISTS. Additionally, we considered models that are based on the principles of Neural ODEs, such as *GRU-ODE* (Brouwer et al. 2019), *ODE-RNN* (Rubanova, Chen, and Duvenaud 2019) and *ODE-LSTM* (Lechner and Hasani 2020). Moreover, recent advancements of Neural CDEs were also included in the evaluation, such as *Neural CDE* (Kidger et al. 2020), *Neural RDE* (Morrill et al. 2021), *ANCDE* (Jhin et al. 2024), *EXIT* (Jhin et al. 2022), *LEAP* (Jhin et al. 2023), and *DualDynamics* (Oh, Lim, and Kim 2025). Lastly, we included naïve *Neural Flow* (Bilos et al. 2021) with different configurations: ResNet, GRU, and Coupling Flow.

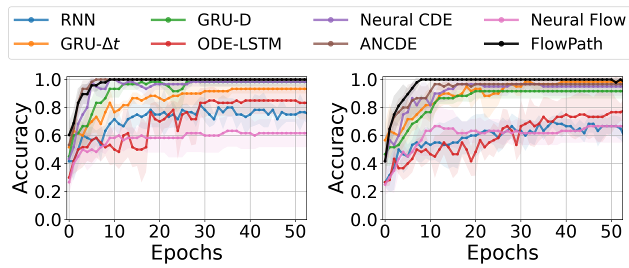
Qualitative Evaluation of Key Properties

We used the 'BasicMotions' dataset for a qualitative analysis. The dataset comprises motion data collected from participants performing four distinct activities while wearing a



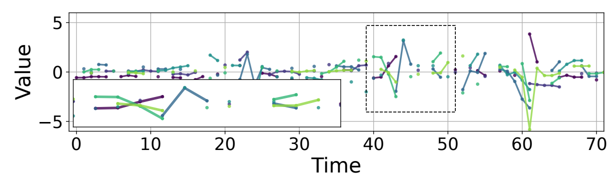
(b) Regular (c) 50% Missing

Figure 1: Test loss curves on 'BasicMotions' under regular and irregular scenarios using selected methods

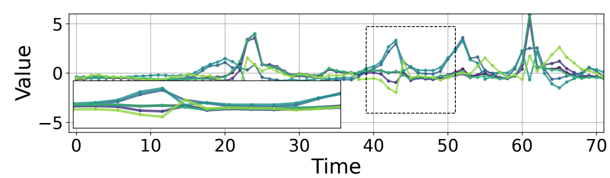


(b) Regular (c) 50% Missing

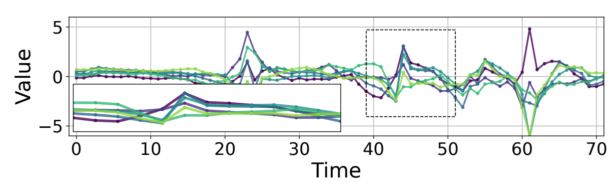
Figure 2: Test accuracy on 'BasicMotions' under regular and irregular scenarios using selected methods and FlowPath



(a) Irregularly-sampled time series (ISTS)



(b) MLP-based non-invertible path



(c) Flow-based invertible path

Figure 3: Qualitative comparison of path construction methods on a sample from the 'BasicMotions' dataset with 50% missingness. (a) Raw and sparse observations. (b) Path generated by a non-invertible MLP, exhibiting a biased curve toward the observed points. (c) Path produced by the proposed FlowPath, capturing a more structured and stable manifold. Each color denotes a separate data dimension.

smartwatch. Each activity was performed five times, with data sampled every 0.1 seconds over a 10-second period.

In Figures 1 and 2, FlowPath consistently achieves the lowest loss and highest accuracy, demonstrating faster and more stable convergence across both regular and missing data scenarios on the ‘BasicMotions‘ dataset. This empirical observation of stable learning and robust performance aligns with FlowPath’s theoretical properties: the well-posedness of its dynamics (Theorem 2) contributes to stable training, while its generalization capability (Theorem 3) suggests that strong empirical performance can translate effectively to unseen test data.

Our goal is not to reconstruct the exact path but to learn a meaningful continuous representation $\Phi(t)$ for classification. Figure 3 shows a qualitative comparison of the manifolds learned from sparse data. Under highly irregular observations (a), a standard non-invertible multi-layer perceptron (MLP) (b) produces a disordered path that overfits sparse points without capturing the underlying structure. In contrast, FlowPath (c) learns a smoother and more coherent path. This stability arises from the invertibility constraint, which promotes information preservation and geometric consistency (Theorem 1). Additional visualizations are provided in the supplementary material.

Structural Comparison of Learned Manifolds

To assess the structure of the learned representations, we analyze trajectories and distributions. For clarity, three dimensions from the observation space are shown here. A full analysis is provided in the supplementary material. Furthermore, we include an extended discussion on NDEs’ robustness and manifold learning.

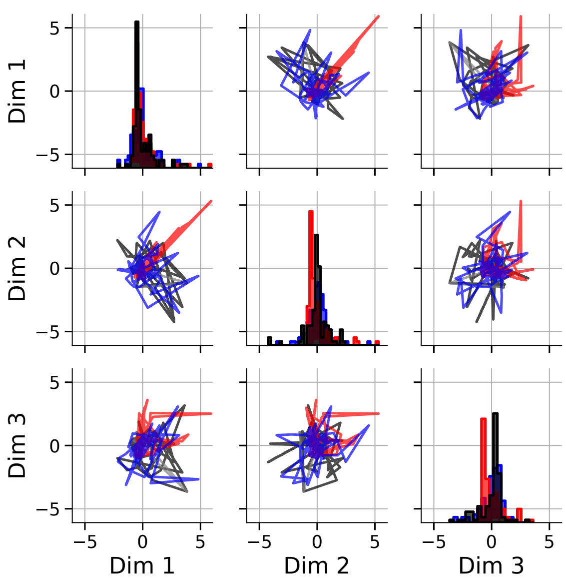


Figure 4: 2D projections of learned trajectories from sparse input. Off-diagonal plots show phase-space paths, and diagonal plots show marginal histograms. FlowPath (blue) better matches the ground truth (black) than the MLP (red).

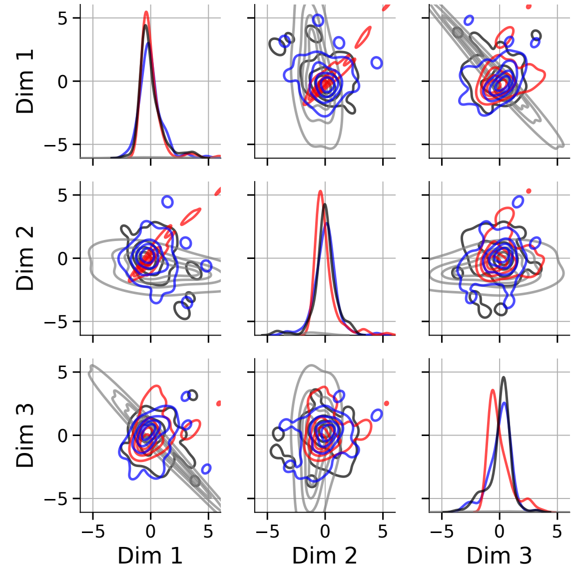
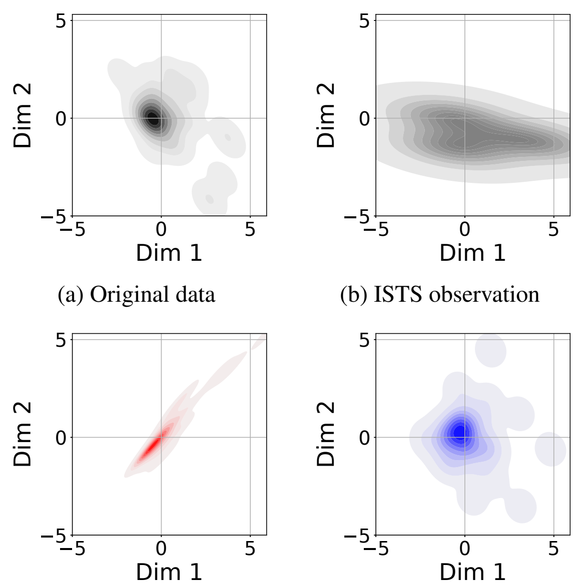


Figure 5: 1D and 2D Kernel Density Estimates (KDEs) for each method. FlowPath (blue) aligns more closely with the original distribution (black) than the MLP (red).



(a) Original data

(b) ISTS observation

(c) MLP-based non-invertible path

(d) Flow-based invertible path

Figure 6: 2D KDEs of the learned manifolds between Dim 1 and Dim 2, using the sparse observation shown in panel (b)

Figure 4 highlights structural differences in the learned manifolds. Sparse observations (gray) lack temporal continuity due to missing data. The MLP path (red) shows instability and deviation from the original geometry (black), likely overfitting to observed points. FlowPath (blue), in contrast, maintains a smoother and more coherent trajectory, reflecting the benefits of the invertibility constraint.

Figure 5 shows KDEs of the learned manifolds. FlowPath achieves better alignment with the true distribution

Methods	Regular		30% Missing		50% Missing		70% Missing		Average	
	Accuracy	Rank	Accuracy	Rank	Accuracy	Rank	Accuracy	Rank	Accuracy	Rank
RNN	0.560 (0.072)	10.7	0.484 (0.075)	13.3	0.471 (0.082)	12.9	0.453 (0.068)	13.3	0.492 (0.074)	12.6
LSTM	0.588 (0.067)	10.0	0.552 (0.075)	9.4	0.516 (0.073)	10.5	0.505 (0.067)	10.6	0.540 (0.071)	10.1
GRU	0.674 (0.080)	6.9	0.639 (0.065)	8.0	0.611 (0.076)	8.2	0.606 (0.088)	8.1	0.633 (0.077)	7.8
GRU-Δt	0.629 (0.065)	9.1	0.636 (0.069)	7.6	0.651 (0.068)	6.7	0.649 (0.074)	7.5	0.641 (0.069)	7.7
GRU-D	0.593 (0.088)	10.0	0.579 (0.087)	9.8	0.580 (0.075)	9.7	0.599 (0.062)	9.4	0.588 (0.078)	9.7
GRU-ODE	0.663 (0.072)	7.2	0.661 (0.069)	6.8	0.664 (0.069)	6.5	0.659 (0.081)	6.3	0.662 (0.073)	6.7
ODE-RNN	0.652 (0.085)	6.8	0.632 (0.076)	7.3	0.626 (0.086)	7.2	0.653 (0.059)	5.8	0.641 (0.076)	6.8
ODE-LSTM	0.566 (0.074)	10.4	0.518 (0.069)	11.4	0.501 (0.068)	12.1	0.474 (0.068)	12.3	0.515 (0.070)	11.5
Neural CDE	0.681 (0.073)	7.1	0.672 (0.068)	7.3	0.661 (0.070)	7.0	0.652 (0.091)	7.0	0.667 (0.075)	7.1
Neural RDE	0.649 (0.082)	7.9	0.648 (0.071)	6.9	0.633 (0.078)	7.7	0.607 (0.079)	8.2	0.634 (0.078)	7.7
ANCDE	0.662 (0.083)	7.3	0.661 (0.083)	6.8	0.639 (0.080)	7.6	0.631 (0.073)	7.1	0.649 (0.080)	7.2
EXIT	0.595 (0.087)	9.4	0.580 (0.088)	9.8	0.578 (0.086)	9.5	0.564 (0.072)	10.0	0.579 (0.083)	9.7
LEAP	0.490 (0.062)	13.1	0.459 (0.070)	13.7	0.466 (0.074)	12.4	0.451 (0.074)	12.7	0.466 (0.070)	13.0
DualDynamics	<u>0.724 (0.090)</u>	<u>4.6</u>	<u>0.720 (0.088)</u>	<u>4.9</u>	<u>0.691 (0.091)</u>	<u>4.9</u>	<u>0.697 (0.098)</u>	<u>4.6</u>	<u>0.708 (0.092)</u>	<u>4.8</u>
Neural Flow	0.530 (0.069)	11.8	0.531 (0.072)	9.9	0.537 (0.073)	9.4	0.535 (0.082)	9.7	0.533 (0.074)	10.2
FlowPath	**0.731 (0.083)**	**3.7**	**0.743 (0.091)**	**3.1**	**0.726 (0.084)**	**3.6**	**0.718 (0.090)**	**3.5**	**0.730 (0.087)**	**3.5**

Table 1: Classification accuracy on 18 benchmark datasets under various scenarios. (Values in parentheses denote the mean standard deviation across different datasets and scenarios. **Best** and <u>second-best</u> results are highlighted, respectively.)

across both marginal and joint spaces. This ability to reconstruct distributional structure under sparsity contributes to its downstream classification performance. To illustrate this in detail, we zoom in on one representative case.

Figure 6 qualitatively analyzes the ability to recover the underlying manifold (a) from only the sparse observations in (b). The non-invertible MLP (c) learns a distorted distribution that is misaligned with the unknown original geometry. In contrast, FlowPath (d) yields a more coherent manifold that better aligns with the original distribution, suggesting that the structural constraint of invertibility is crucial for learning a faithful representation under sparsity.

Performance Comparison with 18 Datasets

In Table 1, we present a comparative analysis of average classification accuracy across various scenarios, including the regular scenario and irregular scenarios (with missing rates of 30%, 50%, and 70%), alongside an aggregate average across all settings. Notably, FlowPath maintains robustness under high rates of missing data, where other methods experience a significant performance drop. Additionally, FlowPath not only achieves the highest average accuracy but also consistently ranks among the top-performing models overall across the four missingness scenarios.

Settings	Regular	30% Missing	50% Missing	70% Missing	All Settings
Wins	8	11	11	10	40
Ties	2	1	2	1	6
Losses	8	6	5	7	26

Table 2: Pairwise comparison (wins / ties / losses) against DualDynamics across different missing rates.

As shown in Table 2, our method outperforms the strongest baseline, DualDynamics, under missing-value conditions. Performance is similar in the regular setting, but our model gains more wins as the missing rate increases, showing greater robustness to incomplete observations. Complete baseline comparisons and pairwise statistical test results are provided in the supplementary material.

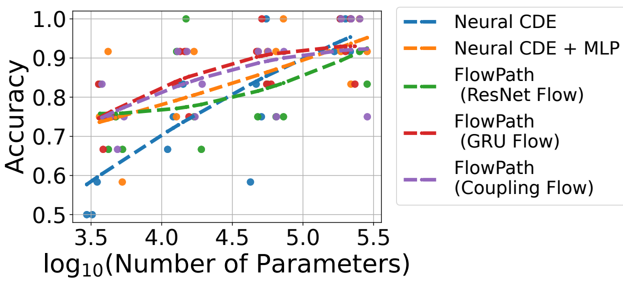
Table 3 presents an ablation study that investigates the sources of FlowPath's performance improvements. The results indicate that while a simple learned path incorporated with an MLP yields modest gains over the fixed-path Neural CDE baseline, the structurally constrained FlowPath consistently achieves the highest performance across all levels of missingness. Furthermore, additional experiments reported in the supplementary material compare three invertible flow architectures, including ResNet, GRU, and Coupling Flow, demonstrating that FlowPath's effectiveness is robust to the choice of flow architecture.

Settings	Regular	30% Missing	50% Missing	70% Missing	All Settings
Neural CDE	0.681 (0.073)	0.672 (0.068)	0.661 (0.070)	0.652 (0.091)	0.667 (0.075)
+ MLP	0.705 (0.077)	0.700 (0.084)	0.695 (0.091)	0.655 (0.084)	0.689 (0.084)
FlowPath	0.731 (0.083)	0.743 (0.091)	0.726 (0.084)	0.718 (0.090)	0.730 (0.087)

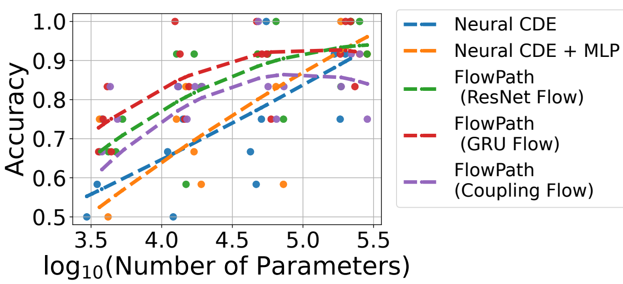
Table 3: Ablation study comparing the standard Neural CDE, a non-invertible MLP, and the proposed FlowPath.

Performance-computation Analysis

We conduct an in-depth comparison across different hyperparameter settings, varying the flow architecture (ResNet, GRU, Coupling Flow), the number of layers n_l, and hidden sizes n_h. Figures 7 (a) and (b) compare classification accuracy to model complexity in the regular and 50% missing settings, respectively. FlowPath variants consistently outperform the Neural CDE baseline at matched parameter counts, with performance gaps widening under higher missingness. MLP-based paths may perform adequately in certain cases; however, their performance significantly deteriorates under irregular scenarios. These results show that FlowPath's invertible, learnable path improves performance with moderate computational cost, making it a practical choice.



(a) Performance on 'Regular' scenario



(b) Performance on '50% Missing' scenario

Figure 7: Analysis of performance-computation trade-off. Figures show $\log_{10}$ of the parameter count versus classification accuracy. Each colored dot corresponds to specific hyperparameters and the dashed line indicates the fitted trend.

Experiment: Real-world HAR Dataset

To evaluate robustness on a real-world task, we used the PAMAP2 dataset (A. Reiss and D. Stricker 2012), a challenging benchmark for sensor-based Human Activity Recognition (HAR). Each participant was equipped with three sensors positioned on the wrist, chest, and ankle, as well as a heart rate monitor. The dataset comprises 5333 segments spanning 8 daily living activities, captured using 17 distinct sensor modalities.

We followed the exact preprocessing protocol, data splits, and sensor dropout methodology proposed by Zhang et al. (2022). To simulate irregularity, we randomly removed 60% of data points from each segment, using the same random mask across all experiments. After that, we generated scenarios by randomly removing a subset of input sensors with dropout rates ranging from 10% to 50%. This approach increases the challenge of the problem, as the omitted sensors are randomly selected for each sample. For every test sample, a subset of sensors was designated as missing, and all observations from these sensors were replaced with zeros.

Benchmark Methods

We compare the proposed FlowPath with the following state-of-the-art baselines, including *Transformer* (Vaswani et al. 2017), *Trans-mean* (Transformer with mean imputation), *GRU-D* (Chen et al. 2018), *SeFT* (Horn et al. 2020), *mTAND* (Shukla and Marlin 2021), *Raindrop* (Zhang et al. 2022), and *Neural CDE* (Kidger et al. 2020).

For the original dataset without sensor dropout, we have included additional benchmarks such as *DGM*^{2}*-O* (Wu et al. 2021), *IP-Net* (Shukla and Marlin 2019), *MTGNN* (Wu et al. 2020), *TITD* (Ji et al. 2025), and *CoFormer* (Y. Wei et al. 2023). For fair comparison, we used performance results as reported in the original literature.

Performance Comparison

As detailed in the supplementary material, optimal hyperparameters were selected via grid search. We employed the standard Neural CDE and compared various flow configurations for the proposed FlowPath. Regardless of the flow configuration, FlowPath consistently demonstrated superior classification performance compared to Neural CDE, as shown in Table 4. Based on hyperparameter tuning, the GRU Flow was selected for this dataset and scenarios.

Sensor Dropout	Neural CDE		FlowPath		
	Spline	**MLP**	**ResNet**	**GRU**	**Coupling**
0%	95.0 ±0.4	93.9 ±0.4	94.6 ±0.5	95.6 ±0.2	94.8 ±0.2
10%	87.3 ±1.3	86.4 ±1.0	87.2 ±0.3	88.3 ±0.4	87.4 ±0.3
20%	76.8 ±0.6	75.1 ±0.8	79.3 ±1.0	79.1 ±1.9	77.8 ±0.6
30%	67.1 ±1.9	68.2 ±0.8	71.4 ±0.7	70.0 ±0.7	70.2 ±1.0
40%	57.7 ±1.6	58.8 ±2.8	65.9 ±1.9	59.6 ±1.0	60.8 ±1.1
50%	51.6 ±1.0	50.7 ±2.0	57.9 ±0.2	54.1 ±1.0	54.2 ±3.1

Table 4: F1 score comparison on the PAMAP2 dataset between Neural CDE and the proposed FlowPath model.

Table 5 compares classification metrics for various methods under the irregularly-sampled PAMAP2 dataset with all sensors. FlowPath's learnable invertible flow further boosts performance, indicating its superior handling of ISTS.

Methods	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Transformer	83.5±1.5	84.8±1.5	86.0±1.2	85.0±1.3
Trans-mean	83.7±2.3	84.9±2.6	86.4±2.1	85.1±2.4
GRU-D	83.3±1.6	84.6±1.2	85.2±1.6	84.8±1.2
SeFT	67.1±2.2	70.0±2.4	68.2±1.5	68.5±1.8
mTAND	74.6±4.3	74.3±4.0	79.5±2.8	76.8±3.4
Raindrop	88.5±1.5	89.9±1.5	89.9±0.6	89.8±1.0
DGM^2-O	82.4±2.3	85.2±1.2	83.9±2.3	84.3±1.8
IP-Net	74.3±3.8	75.6±2.1	77.9±2.2	76.6±2.8
MTGNN	83.4±1.9	85.2±1.7	86.1±1.9	85.9±2.4
TITD	90.7±0.5	90.2±1.3	90.7±0.9	90.5±1.5
CoFormer	91.2±0.6	92.4±0.7	93.7±0.7	92.8±0.5
Neural CDE	94.2±0.5	95.2±0.4	94.8±0.5	95.0±0.4
FlowPath	**94.8±0.2**	**95.8±0.4**	**95.5±0.2**	**95.6±0.2**

Table 5: Classification performance on the PAMAP2 dataset with all sensors under original irregular sampling.

Furthermore, Table 6 summarizes the classification outcomes for the PAMAP2 dataset using a variety of benchmark methods. Across a comparison with all benchmark methods, our approach consistently delivers superior performance across all evaluated metrics.

	Methods	Accuracy	Precision	Recall	F1 Score
10% Missing	Transformer	60.9 ±12.8	58.4 ±18.4	59.1 ±16.2	56.9 ±18.9
	Trans-mean	62.4 ±3.5	59.6 ±7.2	63.7 ±8.1	62.7 ±6.4
	GRU-D	68.4 ±3.7	74.2 ±3.0	70.8 ±4.2	72.0 ±3.7
	SeFT	40.0 ±1.9	40.8 ±3.2	41.0 ±0.7	39.9 ±1.5
	mTAND	53.4 ±2.0	54.8 ±2.7	57.0 ±1.9	55.9 ±2.2
	Raindrop	76.7 ±1.8	79.9 ±1.7	77.9 ±2.3	78.6 ±1.8
	Neural CDE	85.8 ±1.4	88.8 ±1.2	86.3 ±1.3	87.3 ±1.3
	FlowPath	**86.6 ±1.1**	**89.5 ±0.8**	**87.4 ±0.4**	**88.3 ±0.4**
20% Missing	Transformer	62.3 ±11.5	65.9 ±12.7	61.4 ±13.9	61.8 ±15.6
	Trans-mean	56.8 ±4.1	59.4 ±3.4	53.2 ±3.9	55.3 ±3.5
	GRU-D	64.8 ±0.4	69.8 ±0.8	65.8 ±0.5	67.2 ±0.0
	SeFT	34.2 ±2.8	34.9 ±5.2	34.6 ±2.1	33.3 ±2.7
	mTAND	45.6 ±1.6	49.2 ±2.1	49.0 ±1.6	49.0 ±1.0
	Raindrop	71.3 ±2.5	75.8 ±2.2	72.5 ±2.0	73.4 ±2.1
	Neural CDE	74.4 ±0.5	81.0 ±0.8	74.7 ±1.0	76.8 ±0.6
	FlowPath	**77.0 ±1.7**	**82.4 ±1.4**	**77.1 ±2.3**	**79.1 ±1.9**
30% Missing	Transformer	52.0 ±11.9	55.2 ±15.3	50.1 ±13.3	48.4 ±18.2
	Trans-mean	65.1 ±1.9	63.8 ±1.2	**67.9 ±1.8**	64.9 ±1.7
	GRU-D	58.0 ±2.0	63.2 ±1.7	58.2 ±3.1	59.3 ±3.5
	SeFT	31.7 ±1.5	31.0 ±2.7	32.0 ±1.2	28.0 ±1.6
	mTAND	34.7 ±5.5	43.4 ±4.0	36.3 ±4.7	39.5 ±4.4
	Raindrop	60.3 ±3.5	68.1 ±3.1	60.3 ±3.6	61.9 ±3.9
	Neural CDE	65.5 ±1.8	73.1 ±1.8	64.9 ±2.3	67.1 ±1.9
	FlowPath	**67.3 ±0.2**	**75.1 ±0.4**	67.7 ±0.7	**70.0 ±0.7**
40% Missing	Transformer	43.8 ±14.0	44.6 ±23.0	40.5 ±15.9	40.2 ±20.1
	Trans-mean	48.7 ±2.7	55.8 ±2.6	54.2 ±3.0	55.1 ±2.9
	GRU-D	47.7 ±1.4	63.4 ±1.6	44.5 ±0.5	47.5 ±0.0
	SeFT	26.8 ±2.6	24.1 ±3.4	28.0 ±1.2	23.3 ±3.0
	mTAND	23.7 ±1.0	33.9 ±6.5	26.4 ±1.6	29.3 ±1.9
	Raindrop	57.0 ±3.1	65.4 ±2.7	56.7 ±3.1	58.9 ±2.5
	Neural CDE	55.8 ±0.5	**67.7 ±1.5**	54.5 ±2.3	57.7 ±1.6
	FlowPath	**58.1 ±0.1**	67.0 ±0.9	**57.1 ±0.7**	**59.6 ±1.0**
50% Missing	Transformer	43.2 ±2.5	52.0 ±2.5	36.9 ±3.1	41.9 ±3.2
	Trans-mean	46.4 ±1.4	59.1 ±3.2	43.1 ±2.2	46.5 ±3.1
	GRU-D	49.7 ±1.2	52.4 ±0.3	42.5 ±1.7	47.5 ±1.2
	SeFT	26.4 ±1.4	23.0 ±2.9	27.5 ±0.4	23.5 ±1.8
	mTAND	20.9 ±3.1	35.1 ±6.1	23.0 ±3.2	27.7 ±3.9
	Raindrop	47.2 ±4.4	59.4 ±3.9	44.8 ±5.3	47.6 ±5.2
	Neural CDE	49.9 ±1.7	65.5 ±0.8	48.9 ±1.5	51.6 ±1.0
	FlowPath	**52.0 ±0.6**	**66.3 ±1.9**	**51.3 ±0.7**	**54.1 ±1.0**

Table 6: Classification performance on the PAMAP2 dataset with sensor dropout rates ranging from 10% to 50%.

Experiment: Real-world Medical Dataset

We further evaluated model performance on a medical classification benchmark on the PhysioNet Sepsis dataset (Reyna et al. 2020), which includes 40,335 patients and 34 temporal variables. Following Kidger et al. (2020), we compared models with and without observation intensity (OI), an index reflecting patient severity, and measured classification performance by AUROC.

Benchmark models and reported performance from Kidger et al. (2020), Oh, Lim, and Kim (2025), and Oh et al. (2025). We also include variants of stable Neural SDEs (Oh, Lim, and Kim 2024), including Neural Langevin-type SDE (LSDE), Neural Linear Noise SDE (LNSDE), and Neural Geometric SDE (GSDE). For implementation details and benchmark methods, please consult the original studies.

Methods	Test AUROC	
	With OI	Without OI
GRU-Δt	0.878 ±0.006	0.840 ±0.007
GRU-D	0.871 ±0.022	0.850 ±0.013
GRU-ODE	0.852 ±0.010	0.771 ±0.024
ODE-RNN	0.874 ±0.016	0.833 ±0.020
Latent-ODE	0.787 ±0.011	0.495 ±0.002
ACE-NODE	0.804 ±0.010	0.514 ±0.003
Neural CDE	0.880 ±0.006	0.776 ±0.009
ANCDE	0.900 ±0.002	0.823 ±0.003
EXIT	0.913 ±0.002	0.836 ±0.003
DualDynamics	0.918 ±0.003	0.873 ±0.004
Neural SDE	0.799 ±0.007	0.796 ±0.006
Neural LSDE	0.909 ±0.004	0.879 ±0.008
Neural LNSDE	0.911 ±0.002	0.881 ±0.002
Neural GSDE	0.909 ±0.001	0.884 ±0.002
FlowPath		
– ResNet	0.919±0.005	0.869±0.006
– GRU	0.918±0.005	0.870±0.005
– Coupling	0.916±0.006	0.866±0.002

Table 7: AUROC Comparison on PhysioNet Sepsis

As shown in Table 7, FlowPath variants achieve the highest or near-highest AUROC among all baselines. These results suggest that learning continuous flow variations offers a reliable inductive bias for modeling complex temporal dynamics in time series analysis.

Conclusion

We present *FlowPath*, a framework for modeling ISTS by learning the control path of a Neural CDE through an invertible neural flow. This approach replaces fixed interpolation with a data-adaptive, structure-preserving transformation that more accurately captures the underlying temporal geometry. Experimental results on 18 benchmark datasets and a real-world case study demonstrate improved classification accuracy and robustness, particularly under conditions of high missingness. These findings highlight the importance of modeling the geometry of the control path, not just the dynamics along it, to build reliable continuous-time models for real-world applications.

While FlowPath improves robustness by learning a well-structured control path, its use of invertible flows introduces additional parameters. Exploring more efficient architectures could reduce this overhead. Furthermore, our evaluation also focuses on classification, and the performance of FlowPath on tasks such as forecasting and generative modeling remains an important area for future investigation.

Acknowledgments

We thank the teams and individuals for their efforts in the real-world dataset preparation and curation for our research, especially the UEA & UCR repository for the numerous datasets that we extensively analyzed.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2024-00407852); the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(No.RS-2020-II201336, Artificial Intelligence graduate school support(UNIST)); the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25442824, AI STAR Fellowship); the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.RS-2023-00218913, RS-2025-00563597, No. RS-2025-02216640)

References

A. Reiss; and D. Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *2012 16th International Symposium on Wearable Computers*, 108–109.

Bagnall, A.; Dau, H. A.; Lines, J.; Flynn, M.; Large, J.; Bostrom, A.; Southam, P.; and Keogh, E. 2018. The UEA multivariate time series classification archive, 2018.

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. 2017. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6240–6249.

Bilos, M.; Sommer, J.; Rangapuram, S. S.; Januschowski, T.; and Günnemann, S. 2021. Neural Flows: Efficient Alternative to Neural ODEs. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual.*, 21325–21337.

Brouwer, E. D.; Simm, J.; Arany, A.; and Moreau, Y. 2019. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*, 7377–7388.

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, M.; Li, X.; and Zhao, T. 2020. On Generalization Bounds of a Family of Recurrent Neural Networks. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, 1233–1243.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*, 6572–6583.

Choi, E.; Bahadori, M. T.; Schuetz, A.; Stewart, W. F.; and Sun, J. 2016. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. In Doshi-Velez, F.; Fackler, J.; Kale, D.; Wallace, B.; and Wiens, J., eds., *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, 301–318. Northeastern University, Boston, MA, USA: PMLR.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press.

H. A. Dau; A. Bagnall; K. Kamgar; C. -C. M. Yeh; Y. Zhu; S. Gharghabi; C. A. Ratanamahatana; and E. Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6): 1293–1305.

Horn, M.; Moor, M.; Bock, C.; Rieck, B.; and Borgwardt, K. 2020. Set Functions for Time Series. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 4353–4363. Proceedings of Machine Learning Research: PMLR.

Hossain, T.; Ahad, M. A.; and Inoue, S. 2020. A Method for Sensor-Based Activity Recognition in Missing Data Scenario. *Sensors*, 20(14).

Ishikawa, I.; Teshima, T.; Tojo, K.; Oono, K.; Ikeda, M.; and Sugiyama, M. 2023. Universal Approximation Property of Invertible Neural Networks. *Journal of Machine Learning Research*, 24(287): 1–68.

Jhin, S. Y.; Jo, M.; Kook, S.; and Park, N. 2023. Learnable Path in Neural Controlled Differential Equations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7): 8014–8022.

Jhin, S. Y.; Lee, J.; Jo, M.; Kook, S.; Jeon, J.; Hyeong, J.; Kim, J.; and Park, N. 2022. EXIT: Extrapolation and Interpolation-based Neural Controlled Differential Equations for Time-series Classification and Forecasting. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, 3102–3112.

Jhin, S. Y.; Shin, H.; Kim, S.; Hong, S.; Jo, M.; Park, S.; Park, N.; Lee, S.; Maeng, H.; and Jeon, S. 2024. Attentive neural controlled differential equations for time-series classification and forecasting. *Knowledge and Information Systems*, 66(3): 1885–1915.

Ji, J.; Cao, Y.; Ma, Y.; and Yan, J. 2025. TITD: enhancing optimized temporal position encoding with time intervals and temporal decay in irregular time series forecasting. *Applied Intelligence*, 55(6): 415.

Kidger, P. 2022. On Neural Differential Equations.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. J. 2020. Neural Controlled Differential Equations for Irregular Time Series. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

Lechner, M.; and Hasani, R. 2020. Learning Long-Term Dependencies in Irregularly-Sampled Time Series.

Li, Q.; Lin, T.; and Shen, Z. 2022. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, 25(5): 1671–1709.

Lin, H.; and Jegelka, S. 2018. ResNet with one-neuron hidden layers is a Universal Approximator. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*, 6172–6181.

Lu, Y.; Zhong, A.; Li, Q.; and Dong, B. 2018. Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3276–3285. PMLR.

Löning, M.; Bagnall, A.; Ganesh, S.; Kazakov, V.; Lines, J.; and Király, F. J. 2019. sktime: A Unified Interface for Machine Learning with Time Series.

Medsker, L.; and Jain, L. C. 1999. *Recurrent Neural Networks: Design and Applications*. CRC Press.

Morrill, J.; Kidger, P.; Yang, L.; and Lyons, T. 2022. On the Choice of Interpolation Scheme for Neural CDEs. *Transactions on Machine Learning Research*.

Morrill, J.; Salvi, C.; Kidger, P.; and Foster, J. 2021. Neural Rough Differential Equations for Long Time Series. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event.*, 7829–7838.

Oh, Y.; Kam, S.; Lee, J.; Lim, D.-Y.; Kim, S.; and Bui, A. 2025. Comprehensive Review of Neural Differential Equations for Time Series Analysis.

Oh, Y.; Lim, D.; and Kim, S. 2024. Stable Neural Stochastic Differential Equations in Analyzing Irregular Time Series Data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Oh, Y.; Lim, D.-Y.; and Kim, S. 2025. DualDynamics: Synergizing Implicit and Explicit Methods for Robust Irregular Time Series Analysis. In Walsh, T.; Shah, J.; and Kolter, Z., eds., *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, 19730–19739. AAAI Press.

Papamakarios, G.; Nalisnick, E.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2021. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57): 1–64.

Reyna, M. A.; Josef, C. S.; Jeter, R.; Shashikumar, S. P.; Westover, M. B.; Nemati, S.; Clifford, G. D.; and Sharma, A. 2020. Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. *Critical Care Medicine*, 48(2).

Rubanova, Y.; Chen, T. Q.; and Duvenaud, D. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*, 5321–5331.

S. Hochreiter; and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Shukla, S. N.; and Marlin, B. M. 2019. Interpolation-Prediction Networks for Irregularly Sampled Time Series. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Shukla, S. N.; and Marlin, B. M. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Teshima, T.; Ishikawa, I.; Tojo, K.; Oono, K.; Ikeda, M.; and Sugiyama, M. 2020. Coupling-based Invertible Neural Networks Are Universal Diffeomorphism Approximators. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 3362–3373. Curran Associates, Inc.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wu, Y.; Ni, J.; Cheng, W.; Zong, B.; Song, D.; Chen, Z.; Liu, Y.; Zhang, X.; Chen, H.; and Davidson, S. B. 2021. Dynamic Gaussian Mixture based Deep Generative Model For Robust Forecasting on Sparse Multivariate Time Series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1): 651–659.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 753–763. Virtual Event, CA, USA: Association for Computing Machinery.

Y. Wei; J. Peng; T. He; C. Xu; J. Zhang; S. Pan; and S. Chen. 2023. Compatible Transformer for Irregularly Sampled Multivariate Time Series. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1409–1414.

Zhang, X.; Zeman, M.; Tsiligkaridis, T.; and Zitnik, M. 2022. Graph-Guided Network for Irregularly Sampled Multivariate Time Series. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.