# Stochastic Neural Simulator for Generalizing Dynamical Systems across Environments

**Jiaqi Liu**[1,2], **Jiaxu Cui**[1,2*], **Jiayi Yang**[1,2] and **Bo Yang**[1,2*]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China
[2]College of Computer Science and Technology, Jilin University, China
liujq2117@mails.jlu.edu.cn, cjx@jlu.edu.cn, yangjy2117@163.com, ybo@jlu.edu.cn

## Abstract

Neural simulators for modeling complex dynamical systems have been extensively studied for various real-world applications, such as weather forecasting, ocean current prediction, and computational fluid dynamics simulation. Although they have demonstrated powerful fitting and predicting, most existing models are only built to learn single-system dynamics. Several advanced researches have considered learning dynamics across environments, which can exploit the potential commonalities among the dynamics across environments and adapt to new environments. However, these methods still are prone to scarcity problems where per-environment data is sparse or limited. Therefore, we propose a novel CoNDP (Context-Informed Neural ODE Processes) to achieve learning system dynamics from sparse observations across environments. It can fully use contextual information of each environment to better capture the intrinsic commonalities across environments and distinguishable differences among environments while modeling uncertainty of system evolution, producing more accurate predictions. Intensive experiments are conducted on five complex dynamical systems in various fields. Results show that the proposed CoNDP can achieve optimal results compared with common neural simulators and state-of-the-art cross-environmental models.

## 1 Introduction

The learning of dynamical systems is a fundamental task in various scientific domains such as economics [Varian, 1981], geophysics [Samelson and Wiggins, 2006], and epidemiology [Galea *et al.*, 2010]. However, traditional learning methods heavily rely on expert knowledge and may require extensive computational resources, primarily due to the necessity of solving a large number of complex differential equations [Fu *et al.*, 2017; Liu *et al.*, 2021]. A popular current approach is to utilize neural networks to construct data-driven neural simulators [Li *et al.*, 2021; Chen *et al.*, 2018]. These networks can mitigate the need for expert knowledge and enhance computational speed [Li *et al.*, 2021]. They have also been successfully employed to address challenging problems in chaotic dynamics [Linot *et al.*, 2023] and to expedite scientific discovery [Reichstein *et al.*, 2019].

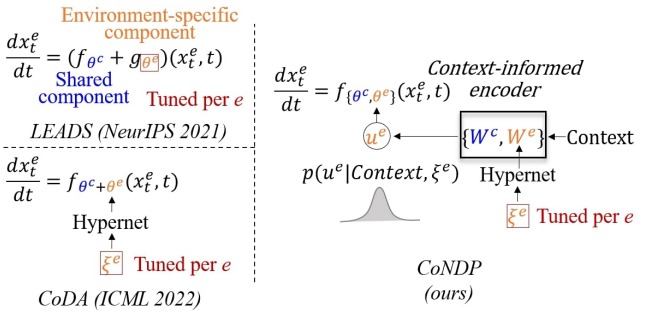



Figure 1: Comparison of state-of-the-art model principles for learning dynamical systems across environments. All models have two components, i.e., a shared component (blue) and an environment-specific component (orange). Both LEADS [Yin *et al.*, 2021] and CoDA [Kirchmeyer *et al.*, 2022] are deterministic. As a comparison, our CoNDP innovatively introduces uncertainty modeling, where environmental parameters are generated by a random control vector ($\boldsymbol{u}^e$), enhancing the model's generalization power.

However, the majority of current neural simulators are constrained by the $i.i.d.$ assumption, which fantasizes that the observed trajectories are abundant and originate from an unchanging environment [Yin *et al.*, 2021]. In reality, data from dynamical systems are influenced by various environmental factors such as gravity, pressure, and temperature [Baradel *et al.*, 2020; Sanchez-Gonzalez *et al.*, 2020], leading to the inadequacy and inefficiency of existing models when applied to real-world data. While it is possible to train multiple models specific to different environments, this approach would require a substantial amount of computational resources and would fail to capture the potential commonalities in dynamics across different environments, resulting in poor predictive performance when the data from individual environments is limited or sparse [Huang *et al.*, 2023]. Consequently, learning dynamical systems across multiple environments remains a fundamental challenge.

Currently, there are several researches exploring cross-

*Corresponding authors

environmental learning models for system dynamics [Yin *et al.*, 2021; Kirchmeyer *et al.*, 2022]. As illustrated in Fig. 1, these models utilize a shared component and an environment-specific component to model ordinary differential equations (ODEs) that characterize system dynamics for each environment, capturing commonalities across environments and the distinct effects of each environment's settings. In practical applications, real-world data may be costly or incomplete due to the expensive and time-consuming nature of data acquisition [Huang *et al.*, 2020], as well as the frequent occurrence of broken sensors or damaged memory units during data collection [Tang *et al.*, 2020]. However, these methods either require numerous parameters to be tuned in the environment-specific component, necessitating a large number of environment-specific trajectories [Yin *et al.*, 2021], or they carry the risk of overfitting in sparse settings [Kirchmeyer *et al.*, 2022]. Moreover, existing methods are deterministic, focusing on learning a deterministic system evolution process, even when dealing with sparse data. This is undoubtedly suboptimal.

To tackle these challenges, we propose a novel Context-informed Neural ODE Processes (CoNDP) to learn dynamical systems from sparse observations across environments. It innovatively introduces uncertainty modeling, where environmental parameters are generated by a random control vector, retaining the advantage in few-shot learning (see Fig. 1). A context-informed encoder is proposed to produce the conditional distribution based on contextual information, i.e., $p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e)$. The incorporation of uncertainty enables the generation of diverse governing equations, thereby enhancing the model's generalization capabilities and supporting learning in sparse settings. Note that the parameters in the encoder are divided into two components, namely $\boldsymbol{W}^c$ and $\boldsymbol{W}^e$, with $\boldsymbol{W}^e$ determined by a hyper network, thereby endowing our model with greater expressive power to capture the inherent commonalities across environments and distinguishable differences among them.

The main contributions are summarized as follows:

- We propose a novel Context-informed Neural ODE Processes (CoNDP) to learn dynamical systems from sparse observations across environments. To our knowledge, this is the first work to incorporate uncertainty into modeling cross-environmental dynamical systems to enhance generalization in sparse setting.
- Extensive experiments are conducted on various complex dynamical systems in ecology, chemistry, and physics, demonstrating the effectiveness of the CoNDP with superior results compared to existing models.

## 2 Related Work and Problem Statement

### 2.1 Related Work

**Learning dynamical systems for a single environment**. Neural simulators modeling system dynamics for a single environment can be broadly categorized into two groups: discrete and continuous. Typical discrete neural simulators are constructed using recurrent neural networks such as LSTM [Hochreiter and Schmidhuber, 1997] and GRU [Cho *et al.*, 2014] due to the power to process sequences. These models often integrate various prior knowledge to enhance learning, such as antisymmetric relations [Chang *et al.*, 2019], system Lagrangians [Rajchakit *et al.*, 2021], and Lyapunov stability [Engelken *et al.*, 2023]. However, prior knowledge requires early efforts or a broad understanding of the dynamical system, which is often impractical or unattainable. Additionally, the discrete nature of these methods makes them struggle with irregularly-sampled observations. Therefore, continuous neural simulators, such as neural ordinary differential equations (neural ODEs) [Chen *et al.*, 2018], have garnered increasing attention. It has been found that neural ODEs can serve as powerful tools for modeling continuous time series [Weerakody *et al.*, 2021; Kidger *et al.*, 2020; Morrill *et al.*, 2021] and system dynamics [Legaard *et al.*, 2023; Böttcher *et al.*, 2022; Linot *et al.*, 2023; Gupta and Lermusiaux, 2021], especially in latent space [Rubanova *et al.*, 2019]. ODE-RNN [Chen *et al.*, 2018] is a representative model that combines neural ODE and recurrent neural networks, where the latter is utilized to update hidden states when observations are available. However, these methods still unrealistically assume that the observed trajectories are abundant and originate from unchanging environments, leading to ill-defined and ineffective modeling for real-world systems.

**Learning dynamical systems across environments**. Currently, there have been several researches on exploring the learning of dynamical systems across different environments [Yin *et al.*, 2021; Kirchmeyer *et al.*, 2022]. LEADS is the first attempt to address multi-environmental scenarios [Yin *et al.*, 2021]. It utilizes a shared component and an environment-specific component to model ODEs that characterize system dynamics in each environment, capturing both commonalities across environments and the distinct effects of each environment's settings as $\frac{\mathrm{d}\boldsymbol{x}_t^e}{\mathrm{d}t} = \big(f_{\boldsymbol{\theta}^c} + g_{\boldsymbol{\theta}^e}\big)(\boldsymbol{x}_t^e, t)$. Here, $\boldsymbol{x}_t^e$ represents the system state at time $t$ in environment $e$, while $\boldsymbol{\theta}^c$ and $\boldsymbol{\theta}^e$ respectively parameterize the commonalities and differences among environments. After the two components are trained across multiple environments, the shared component is frozen, and only the environment-specific component is tuned when adapting to new environments. It directly optimizes $\boldsymbol{\theta^e}$ during adaptation, resulting in the need for a large number of environment-specific trajectories [Yin *et al.*, 2021]. To alleviate the data scale issue, CoDA [Kirchmeyer *et al.*, 2022] introduces a hyper network to produce $\boldsymbol{\theta}^e$. During the adaptation phase, it determines $\boldsymbol{\theta}^e$ by tuning environment-specific parameters $\boldsymbol{\xi}^e$, where $|\boldsymbol{\xi}^e| \ll |\boldsymbol{\theta}^e|$. Consequently, CoDA learns fewer parameters than LEADS during the adaptation phase, mitigating overfitting. However, these methods are still deterministic, focusing on learning a deterministic system evolution process. This makes it difficult to generalize in sparse settings caused by expensive data acquisition [Huang *et al.*, 2020] or malfunctioned sensors [Tang *et al.*, 2020] in the real world. Therefore, we need to consider how to generalize to sparse settings for learning dynamical systems across environments.

### 2.2 Problem Statement

The objective of learning dynamical systems across environments is to construct a generalized neural simulator $S$. This

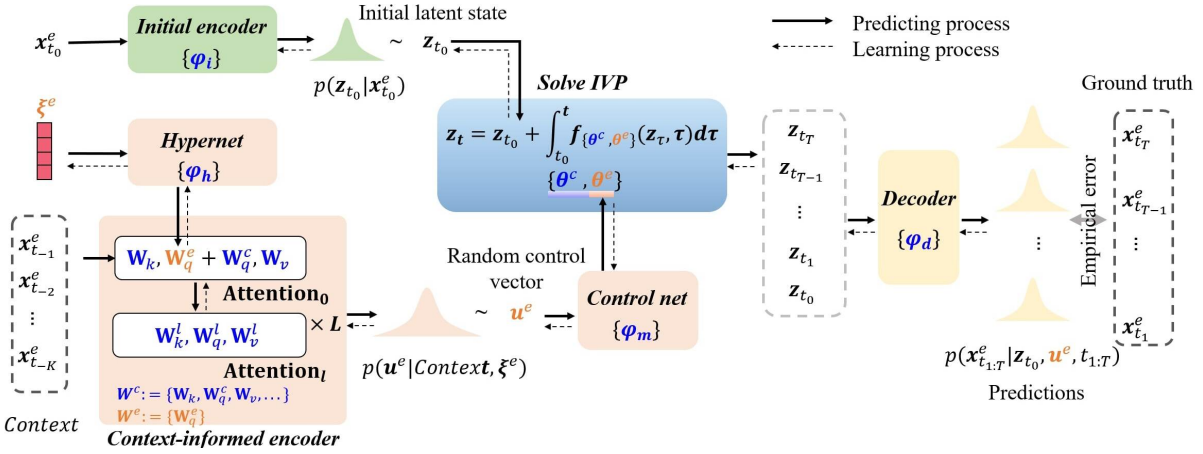



Figure 2: Overview of the CoNDP. Initial encoder estimates the distribution of initial latent state $\boldsymbol{z}_{t_0}$ from $\boldsymbol{x}^e_{t_0}$, i.e., $p(\boldsymbol{z}_{t_0}|\boldsymbol{x}^e_{t_0})$. The context-informed encoder is to map the context into a conditional distribution of environment representation $\boldsymbol{u}^e$, i.e., $p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e)$, which is controlled by the environment-specific parameter $\boldsymbol{\xi}^e$. Environment representation can be seen as a random control vector of system dynamics, which is fed into a control net to produce the environment-specific component of the governing equation $f$. By sampling the initial latent state, we can solve the initial value problem (IVP) to generate the hidden state at any time $t$, i.e., $\boldsymbol{z}_t$. Then, using a decoder to give the predictive distribution.

simulator uses sparse trajectories from the past $K$ timestamps and the current timestamp to predict future trajectories over $1:T$ timestamps for any environment $e$. In other words, it aims to achieve the mapping: $S(\boldsymbol{x}^e_{t_{-K}}, \cdots, \boldsymbol{x}^e_{t_{-1}}, \boldsymbol{x}^e_{t_0}) \rightarrow (\boldsymbol{x}^e_{t_1}, \cdots, \boldsymbol{x}^e_{t_T})$, where the timestamps can have non-uniform intervals and take continuous values.

We consider a dynamical system whose dynamics for environment $e$ is governed by the following ODE:

$$\frac{\mathrm{d}\boldsymbol{x}^e_t}{\mathrm{d}t} = f^e(\boldsymbol{x}^e_t, t), \tag{1}$$

where $\boldsymbol{x}^e_t \in \mathbb{R}^d$ is system state at time $t$ under environment $e \in E$. The dimension of the state is denoted by $d$, and $E$ signifies the set of environments. The observed data from a system under environment $e$ consists of the sparse trajectories $D^e = \{\boldsymbol{x}^e_{t_{-K}}, \boldsymbol{x}^e_{t_{-K+1}}, \cdots, \boldsymbol{x}^e_{t_{-1}}, \boldsymbol{x}^e_{t_0}, \boldsymbol{x}^e_{t_1}, \cdots, \boldsymbol{x}^e_{t_T}\}$. We focus on learning $f^e$ based on the observed data from multiple environments, i.e., $D^1, D^2, \cdots, D^{|E|}$, to capture $f^{e'}$ for any new environment $e'$. This is essential for predicting the system dynamics and achieving a generalized neural simulator. Note that we are considering two tasks: when the new environment $e'$ belongs to the training environment set $E$, we refer to this as the transductive task, whereas when $e'$ is not in $E$, it is referred to as the inductive task.

# 3 CoNDP: Context-Informed Neural ODE Processes

In this section, we present Context-Informed Neural ODE Processes (CoNDP) to generalize to sparse settings for learning dynamical systems across environments.

## 3.1 Model Overview

An overview of the CoNDP is shown in Fig. 2. It consists of four main components: an initial encoder that infers the initial latent states of the system; a context-informed encoder for learning the environment representation, where the attention weights consist of shared weights and environment-specific weights determined by environment-specific parameters and a hyper network; a process of solving initial value problems in hidden space; and a decoder for predicting future system states. Initial encoder estimates the distribution of initial latent state $\boldsymbol{z}_{t_0}$ from initial state $\boldsymbol{x}^e_{t_0}$, i.e., $p(\boldsymbol{z}_{t_0}|\boldsymbol{x}^e_{t_0})$. We denote the historical trajectories over the past $K$ timestamps as $Context = (\boldsymbol{x}^e_{t_{-K}}, \cdots, \boldsymbol{x}^e_{t_{-1}})$. The context-informed encoder is to map the $Context$ into a conditional distribution of environmental representation $\boldsymbol{u}^e$, i.e., $p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e)$, which is determined by environment-specific parameters $\boldsymbol{\xi}^e$ and a hyper network. The environmental representation can be viewed as a random control vector of the system dynamics, which is fed into a control net to produce the environment-specific component of the governing equation $f$. By sampling the initial latent state $\boldsymbol{z}_{t_0}$, we can solve the IVP to generate the hidden state at any time $t$, i.e., $\boldsymbol{z}_t$. Then, a decoder is used to provide the predictive distribution $p(\boldsymbol{x}^e_{1:T}|\boldsymbol{z}_{t_0}, \boldsymbol{u}^e)$. On the contrary, all shared parameters and environment-specific parameters are learned based on the empirical error between predicted and true values.

## 3.2 Model Details

We now introduce the four main components of the model.

**Initial encoder**. Given the initial condition $\boldsymbol{x}^e_{t_0}$ of the dynamical system over environment $e$, the initial encoder $enc_{\boldsymbol{\varphi}_i}$ generates a Gaussian distribution of the initial latent state, i.e., $p(\boldsymbol{z}_{t_0}|\boldsymbol{x}^e_{t_0}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}})$. The mean and variance of the distribution are obtained from

$$\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}^2_{\boldsymbol{z}} = enc_{\boldsymbol{\varphi}_i}(\boldsymbol{x}^e_{t_0}), \tag{2}$$

where $\boldsymbol{\varphi}_i$ is the learnable parameters that are updated during the training phase. The initial latent state $\boldsymbol{z}_{t_0}$ determines the starting point for carrying out dynamic evolution in hidden space. Note that $\boldsymbol{z}_{t_0}$ is a random vector, so it can generate

diverse starting points in prediction. We implement $enc_{\boldsymbol{\varphi}_i}$ using a Multilayer Perception (MLP) whose output is split into two halves to represent the mean and variance respectively.

**Context-informed encoder**. The evolution process of dynamical systems is often influenced by external environments, and environmental characteristics are usually latent and challenging to observe. We thus design an encoder to extract environmental features from data to automatically adapt to the impact of the environment.

As in many cases, the influence of the environment features on system evolution is often manifested in delayed effects rather than instant behaviors [Glass *et al.*, 2021; Smith, 2011; Kuang, 1993; Guglielmi *et al.*, 2022], we employ a temporal self-attention mechanism to learn environment features from observed trajectories by summarizing system behaviors with different delays. This also offers better parallelization for accelerating training speed and alleviates the gradient issues brought by long sequences [Sankar *et al.*, 2020]. Specifically, we use a temporal encoding [Kazemi *et al.*, 2020] to integrate the time delay information.

Given historical trajectories over the past $K$ timestamps under environment $e$ as $Context = (\boldsymbol{x}^e_{t_{-K}}, \cdots, \boldsymbol{x}^e_{t_{-1}})$, the computation of the representation of the observation at time $t_m$ involves conducting a weighted summation over each different time point $t_j$, along with a residual connection. The attention score is determined using a transformer-based approach [Vaswani *et al.*, 2017] and is computed as the dot-product of observation representations derived from value, key, and query projection matrices. We calculate the representation for each historical observation $\boldsymbol{x}^e_{t_m}$, where $m \in \{-1, \cdots, -K\}$, as follows

$$\begin{aligned} \boldsymbol{h}^{(1)}_{t_m} &= \boldsymbol{x}^e_{t_m} + \sigma\left(\sum_{j\neq m} \alpha^{(t_j,t_m)} \times \boldsymbol{W}_v \hat{\boldsymbol{h}}^{(0)}_{t_j}\right), \\ \alpha^{(t_j,t_m)} &= \left(\boldsymbol{W}_k \hat{\boldsymbol{h}}^{(0)}_{t_j}\right)^\top \left((\boldsymbol{W}^c_q + \boldsymbol{W}^e_q)\boldsymbol{x}^e_{t_m}\right) \cdot \frac{1}{\sqrt{d}}, \\ \hat{\boldsymbol{h}}^{(0)}_{t_j} &= \boldsymbol{x}^e_{t_j} + \mathrm{TE}(t_j - t_m), \end{aligned} \tag{3}$$

where $\sigma(\cdot)$ is a non-linear activation function, the temporal encoding $\mathrm{TE}(\cdot)$ satisfying $\mathrm{TE}(\Delta t)_{2i} = \sin\left(\frac{\Delta t}{10000^{2i/d}}\right)$, and $\mathrm{TE}(\Delta t)_{2i+1} = \cos\left(\frac{\Delta t}{10000^{2i/d}}\right)$, $1 \leqslant i \leqslant d$, where $d$ is the state dimension. $\boldsymbol{W}_k$, $\boldsymbol{W}_v$, $\boldsymbol{W}^c_q$, and $\boldsymbol{W}^e_q$ are the parameters in this attention layer. Here we assume the query projection matrix is different among environments by introducing $\boldsymbol{W}^e_q$. This enables our model to receive different attention scores when dealing with different environments, enhancing its ability to capture the intrinsic commonalities and distinguishable differences among environments. $\boldsymbol{W}^e_q$ is determined by a hyper network, with its input being an environment-specific parameter $\boldsymbol{\xi}^e$. This parameter can be viewed as the unobservable implicit characteristic of the environment. As it exclusively focuses on the differences among environments (the commonalities are modeled with other shared parameters in the encoder), $\boldsymbol{\xi}^e$ can be very low-dimensional, enabling rapid adaptation to new environments.

Then, we can obtain the final representation of each observation based on the following attention operations

$$\begin{aligned} \boldsymbol{h}^{(l+1)}_{t_m} &= \boldsymbol{h}^{(l)}_{t_m} + \sigma\left(\sum_{j\neq m} \alpha^{(t_j,t_m)} \times \boldsymbol{W}^l_v \hat{\boldsymbol{h}}^{(l)}_{t_j}\right), \\ \alpha^{(t_j,t_m)} &= \left(\boldsymbol{W}^l_k \hat{\boldsymbol{h}}^{(l)}_{t_j}\right)^\top \left(\boldsymbol{W}^l_q \boldsymbol{h}^{(l)}_{t_m}\right) \cdot \frac{1}{\sqrt{d_h}}, \\ \hat{\boldsymbol{h}}^{(l)}_{t_j} &= \boldsymbol{h}^{(l)}_{t_j} + \mathrm{TE}(t_j - t_m), \end{aligned} \tag{4}$$

where $l = 1, 2, \cdots, L$ and $d_h$ is the dimension of hidden representation. We use the output of the final attention layer as the representation for each observation, denoted as $\boldsymbol{h}^{(L+1)}_{t_m}$.

Following the computation of representations for each observation, we aggregate the entire observed trajectory into a vector to provide conditional distribution of the environment representation, i.e., $p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{u}}, \boldsymbol{\sigma}^2_{\boldsymbol{u}})$, as

$$\boldsymbol{\mu}_{\boldsymbol{u}}, \boldsymbol{\sigma}^2_{\boldsymbol{u}} = enc_{\boldsymbol{\varphi}_u}\left(\frac{1}{K}\sum_{t_m} \sigma\left((\boldsymbol{a}^e)^\top \hat{\boldsymbol{h}}_{t_m} \hat{\boldsymbol{h}}_{t_m}\right)\right), \tag{5}$$

where $\boldsymbol{a}^e = \tanh\left(\left(\frac{1}{K}\sum \hat{\boldsymbol{h}}_{t_m}\right)\boldsymbol{W}_a\right)$ is the average of observation representations with a non-linear transformation and $\hat{\boldsymbol{h}}_{t_m} = \boldsymbol{h}^{(L+1)}_{t_m} + \mathrm{TE}(t)$, $\boldsymbol{W}_a$ is the weights, $K$ is the number of observations for each trajectory, and $enc_{\boldsymbol{\varphi}_u}$ is implemented via the MLP with trainable parameters $\boldsymbol{\varphi}_u$. Note that the environment representation $\boldsymbol{u}^e$ is a random control vector for generating diverse governing equations of dynamical systems, further enhancing the generalization in sparse settings.

**The process of solving initial value problems**. After obtaining two conditional distributions, namely $p(\boldsymbol{z}_{t_0}|\boldsymbol{x}^e_{t_0})$ and $p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e)$, we sample an initial latent state $\boldsymbol{z}_{t_0}$ and an environment representation $\boldsymbol{u}^e$ from these distributions. Subsequently, we can predict the latent states at any time $t$ by solving an initial value problem in the latent space as

$$\boldsymbol{z}_t = \boldsymbol{z}_{t_0} + \int_{t_0}^{t} f_{\{\boldsymbol{\theta}^c, \boldsymbol{\theta}^e\}}(\boldsymbol{z}_\tau, \tau) d\tau, \tag{6}$$

where $\boldsymbol{\theta}^c$ is the shared parameters across environments and $\boldsymbol{\theta}^e$ is environment-specific parameters generated by a control network with parameters $\boldsymbol{\varphi}_m$, i.e., $\boldsymbol{\theta}^e = con_{\boldsymbol{\varphi}_m}(\boldsymbol{u}^e)$.

Due to the sparsity of observed data, it becomes challenging to uniquely determine a dynamical equation and ensure that the data reveals all (or at least most) of the features of the environment. However, existing methods typically only learn a deterministic dynamical process from observed data [Yin *et al.*, 2021; Kirchmeyer *et al.*, 2022] and are unable to capture the uncertainty and multiple solutions in the systems caused by low-resource data. Fortunately, uncertainty modeling can alleviate this issue [Norcliffe *et al.*, 2020]. This is why we use random vectors to model initial signals and environment representations.

**State decoder**. Using latent states $\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \cdots, \boldsymbol{z}_{t_T}$ and placing a Gaussian distribution on system states, we can obtain the predicted system states by employing a decoder as

$$\boldsymbol{\mu}_{\boldsymbol{x}_{t_j}}, \boldsymbol{\sigma}^2_{\boldsymbol{x}_{t_j}} = dec_{\boldsymbol{\varphi}_d}(\boldsymbol{z}_{t_j}), \tag{7}$$

where, $j = 1, 2, \cdots, T$ and $\boldsymbol{\mu}_{\boldsymbol{x}_{t_j}}$ and $\boldsymbol{\sigma}^2_{\boldsymbol{x}_{t_j}}$ are the mean and variance of the predictive distribution of system state, i.e.,

| | Transductive | | | | | Inductive | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LV ($\times 10^{-5}$) | GO ($\times 10^{-4}$) | GS ($\times 10^{-4}$) | NS ($\times 10^{-4}$) | HEAT ($\times 10^{-2}$) | LV ($\times 10^{-5}$) | GO ($\times 10^{-4}$) | GS ($\times 10^{-3}$) | NS ($\times 10^{-4}$) | HEAT ($\times 10^{-2}$) |
| GRU | 7.56 | 141.52 | 8.41 | 173.70 | 367.66 | 274.37 | 7312.40 | 64.62 | 121.24 | 274.42 |
| LSTM | 7.54 | 132.58 | 8.18 | 167.39 | 428.03 | 256.79 | 7286.87 | 70.69 | 118.01 | 274.47 |
| Neural ODE | 8.01 | 130.43 | 7.84 | 169.44 | 417.31 | 299.41 | 7435.18 | 64.61 | 118.19 | 278.50 |
| ODE-RNN | 7.66 | 138.24 | 8.45 | 170.11 | 384.97 | 282.75 | 7951.86 | 63.58 | 115.20 | 293.72 |
| LEADS | 3.75 | 49.96 | 3.34 | 35.23 | 113.30 | 85.12 | 212.31 | 2.46 | 33.02 | 175.49 |
| CoDA | 1.69 | 2.58 | 1.55 | 9.87 | 5.07 | 2.26 | 5.72 | 1.83 | 10.70 | 7.21 |
| CoNDP | **1.54** $\downarrow 8.88\%$ | **2.41** $\downarrow 6.59\%$ | **1.48** $\downarrow 4.52\%$ | **9.33** $\downarrow 5.47\%$ | **4.74** $\downarrow 5.95\%$ | **2.16** $\downarrow 4.42\%$ | **5.54** $\downarrow 3.15\%$ | **1.73** $\downarrow 5.46\%$ | **10.02** $\downarrow 6.36\%$ | **6.98** $\downarrow 3.19\%$ |

Table 1: The average of Mean Square Error (MSE) between predictions and ground truth from various dynamical systems across all testing environments and sparsity levels for both transductive and inductive tasks. The best results are bolded.

$p(\boldsymbol{x}^e_{1:T}|\boldsymbol{z}_{t_0}, \boldsymbol{u}^e, t_{1:T}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}_{t_{1:T}}}, \boldsymbol{\sigma}^2_{\boldsymbol{x}_{t_{1:T}}})$. This predictive distribution can accommodate system noise.

### 3.3 CoNDP as Stochastic Process

From a probabilistic perspective, our proposed CoNDP is a type of neural network-parameterized stochastic process for system states. The Kolmogorov Extension Theorem states that exchangeability and consistency conditions are sufficient to define a stochastic process [Oksendal, 2013]. We demonstrate that the CoNDP satisfies these conditions and present the following proposition.

**Proposition 1** *CoNDP satisfies the exchangeability and consistency conditions.*

In other words, the stochastic processes we have established exist, and the CoNDP is its family of finite-dimensional distributions. As a stochastic process, it should theoretically have excellent generalization power for sparse settings. The detailed proof can be found in supplementary material[1].

### 3.4 Training and Adpating

Now, we introduce the overall training procedure of the CoNDP. We randomly generate multiple environments $E = \{e_1, e_2, ...\}$ for a dynamical system and obtain sparse trajectories through irregularly sampling observations under each environment. These trajectories are divided into two halves along the time, where the first half is $[t_{-K}, t_{-1}]$ for learning environment representation, and the second half, i.e., $[t_0, t_T]$, is to predict system states. We denote the collected data as $D^e = \{\boldsymbol{x}^e_{t_{-K}}, \boldsymbol{x}^e_{t_{-K+1}}, \cdots, \boldsymbol{x}^e_{t_{-1}}, \boldsymbol{x}^e_{t_0}, \boldsymbol{x}^e_{t_1}, \cdots, \boldsymbol{x}^e_{t_T}\}$ for environment $e$. Based on observed data from multiple environments, i.e., $D^{e_1}, D^{e_2}, ...$, we jointly train all parts in an end-to-end way to achieve a generalized neural simulator. Specifically, following [Norcliffe *et al.*, 2020], we learn the model by maximizing the following variational lower bound

$$\sum_{e=e_1,e_2,\ldots} \mathbb{E}_{q_{\boldsymbol{z}_{t_0},\boldsymbol{u}^e}} \Bigg[ \sum_{0 \leqslant i \leqslant T} \log p(\boldsymbol{x}^e_{t_i}|\boldsymbol{z}_{t_0}, \boldsymbol{u}^e, t_i) + \log \frac{p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e)}{p(\boldsymbol{u}^e|D^e \setminus Context, \boldsymbol{\xi}^e)} \Bigg], \tag{8}$$

where $q_{\boldsymbol{z}_{t_0},\boldsymbol{u}^e} = p(\boldsymbol{z}_{t_0}|\boldsymbol{x}^e_{t_0}) p(\boldsymbol{u}^e|Context, \boldsymbol{\xi}^e)$. Detailed implementations, such as neural network architectures, can be found in supplementary material.

For adaptation, we assume that a small number of sparse trajectories are observed under the new environment $e'$ (not more than 10 in experiments). To adapt to the new environment, we freeze all the parameters of the model except the environment-specific parameter $\boldsymbol{\xi}^e$. Using the same loss function as in the training procedure, we only tune the $\boldsymbol{\xi}^e$ in the adaptation phase. After convergence, the entire model is frozen and deployed for testing in the new environment.

## 4 Experiments

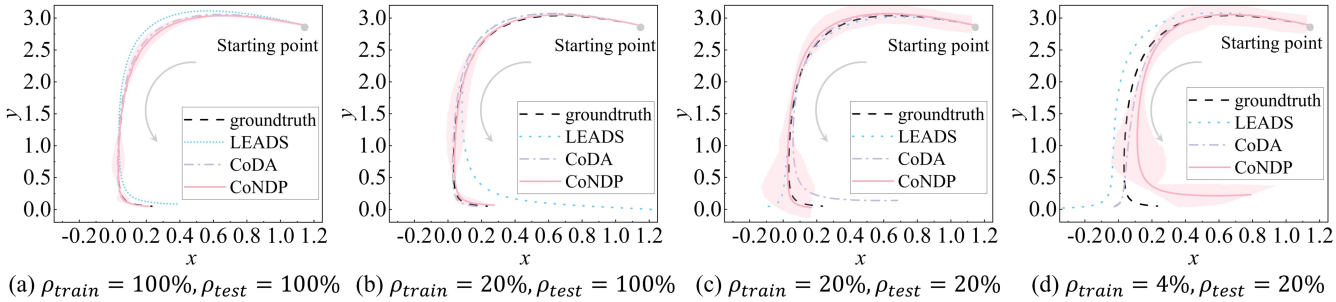
We test the CoNDP[2] on five complex dynamical systems in various fields to answer the following questions: 1) Can the introduction of uncertainty improve model performance, especially in the sparse setting? 2) What factors in our model have an impact on performance? 3) Has our model learned appropriate environment representations and reasonable governing laws of dynamical systems?

**Dynamical systems**. We consider several complex dynamical systems: Lotka-Volterra (LV) predator-prey system [Lotka, 1925] is to describe the dynamics of biological systems where two species interact. Glycolitic-Oscillator (GO) system [Ruoff *et al.*, 2003] is to describe the dynamics of yeast glycolysis in biology. Gray-Scott (GS) [Pearson, 1993] is a reaction-diffusion system where two chemicals react while spreading over space, forming Turing patterns [Horváth *et al.*, 2009]. Navier-Stokes equation (NS) is a fluid dynamics equation [Constantin and Foiaş, 1988] describing the waveform of the fluid system. The heat diffusion equation (HEAT) describes the diffusion of thermal energy [Widder, 1976]. Different parameters in dynamical systems correspond to different environments.

**Building training, tuning, and testing sets**. For the training set, we sample $N^{env}_{tr}$ environments per system and $N^{tra}_{tr}$ trajectories per environment. In addition, we sample $N^{tra}_{tu}$ (up to 10) and $N^{tra}_{te}$ ($\sim 100$) trajectories per new environment for tuning and testing, respectively. A large number of environments from each system are collected as testing environments. If the testing environments have already been

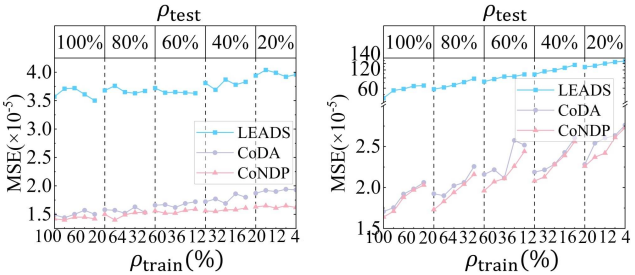
[1] https://github.com/ljqjlu/CoNDP/blob/main/supp-material.pdf

[2] Our code is available at https://github.com/ljqjlu/CoNDP.



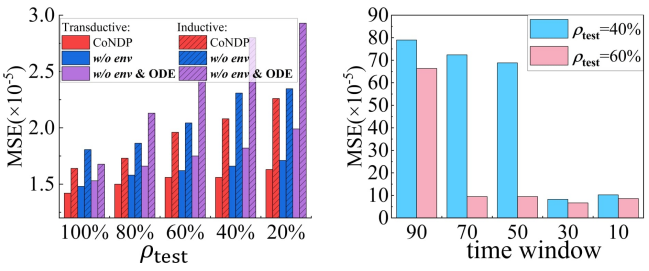

(a) $\rho_{train} = 100\%, \rho_{test} = 100\%$ (b) $\rho_{train} = 20\%, \rho_{test} = 100\%$ (c) $\rho_{train} = 20\%, \rho_{test} = 20\%$ (d) $\rho_{train} = 4\%, \rho_{test} = 20\%$

Figure 3: Comparison of predictive trajectories for a sampled unseen environment of Lotka-Volterra (LV) system under various sparsity levels. $\rho_{train}$ and $\rho_{test}$ denote the ratio of observed data in the training and testing sets, respectively. Due to the uncertainty of our model, the solid line of our model is the mean of predictive trajectories, while the shaded area is twice the standard deviation approximating a 95% confidence interval. It can be seen that our CoNDP performs best even with few observations, and as the number of observations decreases, the uncertainty of the model increases, partially covering the truth.




(a) Transductive (b) Inductive

Figure 4: Comparison of prediction errors under different sparsity levels on Lotka-Volterra (LV) system.




(a) Ablation study (b) Sensitivity analysis

Figure 5: Ablation study and sensitivity analysis for CoNDP.

encountered during training, the task of predicting trajectories becomes a transductive task. Otherwise, if the testing environments are entirely new and outside the scope of the training set, it becomes an inductive task. To evaluate the model's performance in a sparse setting, we randomly drop a fraction of environments and observed points of trajectories for the training, tuning, and testing sets. Additional details regarding the dynamical systems and experimental settings can be found in the supplementary material.

**Baselines**. We compare our model against several representative neural simulators for a single environment, including discrete models such as GRU [Cho *et al.*, 2014] and LSTM [Hochreiter and Schmidhuber, 1997], as well as continuous models such as Neural ODE [Chen *et al.*, 2018] and ODE-RNN [Chen *et al.*, 2018]. Additionally, we compare with state-of-the-art models for learning dynamical systems across environments, namely LEADS [Yin *et al.*, 2021] and CoDA [Kirchmeyer *et al.*, 2022]. Note that both are unable to utilize contextual historical trajectories during the testing phase, but make predictions directly from time $t_0$. Therefore, we fed the context into their tuning phase to ensure fairness.

## 4.1 Performance Evaluation under Sparse Settings

We conducted tests on five systems under different testing environments and sparsity levels. Specifically, for each type of task per system, we tested on 100 environments with the ratio of observed data ranging from 4% to 100%. From Table 1, it is evident that the models across environments significantly outperform the single environment models in all dynamical systems. This demonstrates that the former can effectively distinguish and utilize the commonalities across environments and environment-specific differences, leading to better performance in sparse settings, while the latter suffers from the $i.i.d.$ assumption. Thanks to uncertainty modeling and the introduction of contextual information, our CoNDP achieves optimal results, reducing prediction errors by 3.15% to 8.88% compared to the second-best method. The performance on the inductive task is generally worse than that on the transductive task, which also indicates that adapting to completely new environments in sparse settings is difficult. However, our model still demonstrates generalization power for unseen environments. Fig. 3 provides a visualization of the predictive trajectories for a sampled unseen environment of the LV system under various sparsity levels. We also present predictive errors at different sparsity levels in Fig. 4. The sparse scenarios do indeed weaken model performance. Sufficient training data enables the models to learn the commonalities and differences among environments, thereby improving performance. And, the sparsity of testing data makes it difficult to capture features for new environments. Our CoNDP outperforms both LEADS and CoDA, and the gap becomes increasingly apparent as the setting becomes sparser. The results for other systems are similar and can be found in supplementary material.

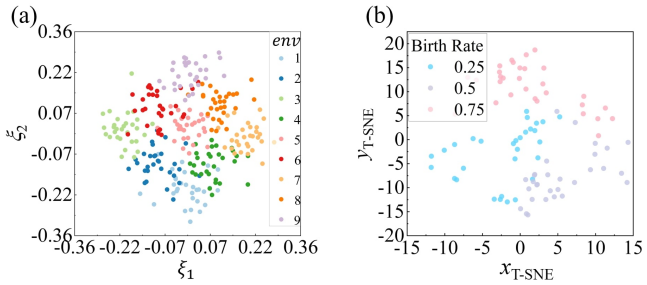



Figure 6: T-SNE visualizations of the environment-specific parameter $\boldsymbol{\xi}^e$ from LV system (a) and the environment representations $\boldsymbol{u}^e$ also from LV system (b).

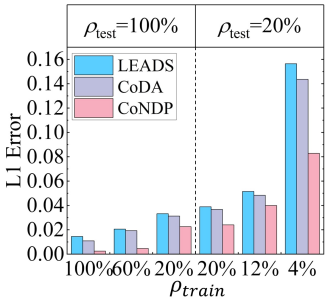



Figure 7: The average L1 error between the coefficients of the discovered governing law and those of the true equation of LV system. The lower the coefficient error, the closer the discovered equation is to the truth.

### 4.2 Ablation Study and Sensitivity Analysis

To analyze the rationality behind our design, we conducted two variants of the CoNDP. One variant involved our model without the environment-specific attention weight $\boldsymbol{W}_q^e$, resulting in the removal of the hyper-network and making the attention weights shared across all environments. The other variant was our model without both the environment-specific component and the ODE process in latent space, directly using neural networks to calculate solutions while bypassing the integration process. These two variants can be viewed as modified versions of Neural Processes [Garnelo *et al.*, 2018] and Neural ODE Processes [Norcliffe *et al.*, 2020], respectively. We tested them on the LV system. From Fig. 5(a), it is evident that our CoNDP outperforms the variants under various levels of sparsity, demonstrating the effectiveness of the design of the context-informed encoder and the ODE component. We also tested the effect of different sizes of context for our model and provided the prediction error for the inductive task on the LV system in Fig. 5(b). Although intuitively, a longer length of context would better reflect the features of the trajectories in the environments, we found that sparsity also affects model performance, with large and irregularly-sampled intervals making it difficult to obtain obvious features. Empirically, we observed that model performance was already satisfactory when the context size was around 30. Thus, the size of the context was set to 30 in our experiments.

### 4.3 Environment Representations

To confirm whether the context-informed encoder can acquire reasonable environment representations and better distinguish the inherent commonalities and differences among environments, we visualize the optimized environment-specific parameter $\boldsymbol{\xi}^e$ and the learned environment representation $\boldsymbol{u}^e$ for a variety of new testing environments by two-dimensional projection. In terms of the optimized $\boldsymbol{\xi}^e$, we observe that they are distinguishable under different environments and are close to each other for similar environments, as shown in Fig. 6(a). For the learned $\boldsymbol{u}^e$, we conducted tests on the Gray-Scott equations and only allowed the reaction rate of one chemical to vary to form new environments. From Fig. 6(b), we see that they are clustered within three reaction rates, indicating that our model can effectively distinguish different environments from the observed trajectories.

### 4.4 Discovering Governing Law of Systems

To evaluate the capability of the CoNDP in capturing the fundamental governing law of the system, we employed a popular sparse symbolic regression, namely SINDy [Brunton *et al.*, 2016], to discover the governing law from the predicted trajectories in unseen environments. However, the relatively large and irregularly-sampled intervals in our settings posed a significant challenge for symbolic regression techniques when using the finite difference method [Perrone and Kao, 1975] to approximate temporal derivatives. We, thus, generated densely-sampled observations from the predictive trajectories to enable the discovery of the governing law, which also serves to validate our model's capacity for temporal hyper-resolution sampling. As depicted in Fig. 7, it becomes difficult to discover the governing law of the system when fewer observations are provided. Our CoNDP demonstrates the ability to discover more accurate equations compared to others, particularly in sparse settings. The discovered equations under different sparsity levels can be found in supplementary material. This reveals that the adaptive attention and probabilistic modeling offer greater flexibility in capturing the essential governing law from limited observations.

## 5 Conclusion

In this study, we propose a novel stochastic neural simulator (CoNDP) designed to learn dynamical systems across environments, thereby overcoming the unrealistic $i.i.d.$ assumption on environments. Comprehensive experiments have demonstrated that it outperforms several representative models designed for a single environment, as well as state-of-the-art models focused on learning across environments. Our analysis reveals that our model excels in distinguishing the inherent commonalities and differences among environments, and the introduction of uncertainty enhances generalization in sparse settings. Nevertheless, challenges persist in effectively modeling counterfactual physical systems and integrating system control for decision-making support. Additionally, the influence of different environments also exists in the network dynamics of the real world. Addressing how to extend the model to high-dimensional network dynamics modeling is an important topic for future research.

## Acknowledgments

This work was supported by the National Science and Technology Major Project under Grant No. 2021ZD0112500; the National Natural Science Foundation of China under Grant Nos. U22A2098, 62172185, 62206105 and 62202200; the Jilin Province Youth Science and Technology Talent Support Project under Grant No. QT202225; the Key Science and Technology Development Plan of Jilin Province under Grant No. 20240302078GX.

## References

[Baradel *et al.*, 2020] Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. Cophy: Counterfactual learning of physical dynamics. In *International Conference on Learning Representations*, 2020.

[Böttcher *et al.*, 2022] Lucas Böttcher, Nino Antulov-Fantulin, and Thomas Asikis. Ai pontryagin or how artificial neural networks learn to control dynamical systems. *Nature communications*, 13(1):333, 2022.

[Brunton *et al.*, 2016] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[Chang *et al.*, 2019] Bo Chang, Minmin Chen, Eldad Haber, and Ed H. Chi. AntisymmetricRNN: A dynamical system view on recurrent neural networks. In *International Conference on Learning Representations*, 2019.

[Chen *et al.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1724. Association for Computational Linguistics, 2014.

[Constantin and Foiaş, 1988] Peter Constantin and Ciprian Foiaş. *Navier-stokes equations*. University of Chicago press, 1988.

[Engelken *et al.*, 2023] Rainer Engelken, Fred Wolf, and Larry F Abbott. Lyapunov spectra of chaotic recurrent neural networks. *Physical Review Research*, 5(4):043044, 2023.

[Fu *et al.*, 2017] Haohuan Fu, Conghui He, Bingwei Chen, Zekun Yin, Zhenguo Zhang, Wenqiang Zhang, Tingjian Zhang, Wei Xue, Weiguo Liu, Wanwang Yin, et al. 9-pflops nonlinear earthquake simulation on sunway taihulight: enabling depiction of 18-hz and 8-meter scenarios. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2017.

[Galea *et al.*, 2010] Sandro Galea, Matthew Riddle, and George A Kaplan. Causal thinking and complex system approaches in epidemiology. *International journal of epidemiology*, 39(1):97–106, 2010.

[Garnelo *et al.*, 2018] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. In *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[Glass *et al.*, 2021] David S Glass, Xiaofan Jin, and Ingmar H Riedel-Kruse. Nonlinear delay differential equations and their application to modeling biological network motifs. *Nature communications*, 12(1):1788, 2021.

[Guglielmi *et al.*, 2022] Nicola Guglielmi, Elisa Iacomini, and Alex Viguerie. Delay differential equations for the spatially resolved simulation of epidemics with specific application to covid-19. *Mathematical Methods in the Applied Sciences*, 45(8):4752–4771, 2022.

[Gupta and Lermusiaux, 2021] Abhinav Gupta and Pierre FJ Lermusiaux. Neural closure models for dynamical systems. *Proceedings of the Royal Society A*, 477(2252):20201004, 2021.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Horváth *et al.*, 2009] Judit Horváth, István Szalai, and Patrick De Kepper. An experimental design method leading to chemical turing patterns. *Science*, 324(5928):772–775, 2009.

[Huang *et al.*, 2020] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems*, 33:16177–16187, 2020.

[Huang *et al.*, 2023] Zijie Huang, Yizhou Sun, and Wei Wang. Generalizing graph ode for learning complex system dynamics across environments. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 798–809, 2023.

[Kazemi *et al.*, 2020] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2020.

[Kidger *et al.*, 2020] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.

[Kirchmeyer *et al.*, 2022] Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to new physical systems via context-informed dynamics model. In *International Conference on Machine Learning*, pages 11283–11301. PMLR, 2022.

[Kuang, 1993] Yang Kuang. *Delay differential equations: with applications in population dynamics*. Academic press, 1993.

[Legaard *et al.*, 2023] Christian Legaard, Thomas Schranz, Gerald Schweiger, Ján Drgoňa, Basak Falay, Cláudio Gomes, Alexandros Iosifidis, Mahdi Abkar, and Peter Larsen. Constructing neural network based models for simulating dynamical systems. *ACM Computing Surveys*, 55(11):1–34, 2023.

[Li *et al.*, 2021] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[Linot *et al.*, 2023] Alec J Linot, Joshua W Burby, Qi Tang, Prasanna Balaprakash, Michael D Graham, and Romit Maulik. Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems. *Journal of Computational Physics*, 474:111838, 2023.

[Liu *et al.*, 2021] Yong Liu, Xin Liu, Fang Li, Haohuan Fu, Yuling Yang, Jiawei Song, Pengpeng Zhao, Zhen Wang, Dajia Peng, Huarong Chen, et al. Closing the" quantum supremacy" gap: achieving real-time simulation of a random quantum circuit using a new sunway supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2021.

[Lotka, 1925] Alfred James Lotka. *Elements of physical biology*. Williams & Wilkins, 1925.

[Morrill *et al.*, 2021] James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pages 7829–7838. PMLR, 2021.

[Norcliffe *et al.*, 2020] Alexander Norcliffe, Cristian Bodnar, Ben Day, Jacob Moss, and Pietro Liò. Neural ode processes. In *International Conference on Learning Representations*, 2020.

[Oksendal, 2013] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[Pearson, 1993] John E Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.

[Perrone and Kao, 1975] Nicholas Perrone and Robert Kao. A general finite difference method for arbitrary meshes. *Computers & Structures*, 5(1):45–57, 1975.

[Rajchakit *et al.*, 2021] G Rajchakit, R Sriraman, N Boonsatit, P Hammachukiattikul, Chee Peng Lim, and P Agarwal. Exponential stability in the lagrange sense for clifford-valued recurrent neural networks with time delays. *Advances in Difference Equations*, 2021(1):1–21, 2021.

[Reichstein *et al.*, 2019] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and fnm Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

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

[Ruoff *et al.*, 2003] Peter Ruoff, Melinda K Christensen, Jana Wolf, and Reinhart Heinrich. Temperature dependency and temperature compensation in a model of yeast glycolytic oscillations. *Biophysical chemistry*, 106(2):179–192, 2003.

[Samelson and Wiggins, 2006] Roger M Samelson and Stephen Wiggins. *Lagrangian transport in geophysical jets and waves: The dynamical systems approach*, volume 31. Springer Science & Business Media, 2006.

[Sanchez-Gonzalez *et al.*, 2020] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.

[Sankar *et al.*, 2020] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pages 519–527, 2020.

[Smith, 2011] Hal L Smith. *An introduction to delay differential equations with applications to the life sciences*, volume 57. springer New York, 2011.

[Tang *et al.*, 2020] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5956–5963, 2020.

[Varian, 1981] Hal R Varian. Dynamical systems with applications to economics. *Handbook of mathematical economics*, 1:93–110, 1981.

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

[Weerakody *et al.*, 2021] Philip B Weerakody, Kok Wai Wong, Guanjin Wang, and Wendell Ela. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178, 2021.

[Widder, 1976] David Vernon Widder. *The heat equation*, volume 67. Academic Press, 1976.

[Yin *et al.*, 2021] Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. Leads: Learning dynamical systems that generalize across environments. *Advances in Neural Information Processing Systems*, 34:7561–7573, 2021.