Learning Slow and Fast System Dynamics via Automatic Separation of Time Scales

Ruikun Li*
Department of Electronic Engineering
BNRist, Tsinghua University
Beijing, China

Huandong Wang†
Department of Electronic Engineering
BNRist, Tsinghua University
Beijing, China

Yong Li
Department of Electronic Engineering
BNRist, Tsinghua University
Beijing, China

ABSTRACT

Learning the underlying slow and fast dynamics of a system is instrumental for many practical applications related to the system. However, existing approaches are limited in discovering the appropriate time scale to separate the slow and fast variables and effectively learning their dynamics based on correct-dimensional representation vectors. In this paper, we introduce a framework that effectively learns slow and fast system dynamics in an integrated manner. We propose a novel intrinsic dimensionality (ID) driven learning method based on a time-lagged autoencoder framework to identify appropriate time scales to separate slow and fast variables and their IDs simultaneously. Further, we propose an integrated framework to concurrently learn the system's slow and fast dynamics, which is able to integrate prior knowledge of time scale and IDs and model the complex coupled slow and fast variables. Extensive experimental results on two representative dynamical systems show that our proposed framework is able to efficiently learn slow and fast system dynamics. Specifically, the long-time prediction performance is able to be improved by 36% on average compared with four representative baselines based on our proposed framework. Furthermore, our proposed system is able to extract interpretable slow and fast dynamics highly correlated with the known slow and fast variables in the dynamical systems. Our codes and datasets are open-sourced at: https://github.com/tsinghua-fib-lab/SlowFastSeparation.

CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; *Learning latent representations*; • **Applied computing** → *Physics.*

KEYWORDS

Dynamical system, slow and fast variables, Koopman theory, time scale

ACM Reference Format:
Ruikun Li, Huandong Wang, and Yong Li. 2023. Learning Slow and Fast System Dynamics via Automatic Separation of Time Scales. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599858

1 INTRODUCTION

The evolution of real-world systems in many scientific fields often involves multiple time scales. For example, in chemical reaction systems there usually exist violent and slow reactions simultaneously [23], where the violent reaction dominates the initial dynamics of the system and quickly reaches a pseudo-steady state, while the slow reaction often dominates the system dynamics after a long-term evolution [17]. By explicitly modeling the violent and slow reactions, we are able to reliably forecast both short-term and long-term states of the system. In a bio-physical system composed of R atoms, it has been found that its effective dimensionality is far smaller than $3R$, which is the degree of freedom of the spatial coordinates of all atoms. Furthermore, the effective dimensionality can be extracted by separation of time scales, where the extracted slow variables are found to characterize more important dynamics and related to the smooth underlying free energy plane [10]. Therefore, modeling slow and fast dynamics is crucial for understanding the underlying structure of the system. In modeling molecular dynamics, extracting the slow collective variables of the system is also an irreplaceable technical foundation [22]. Overall, learning the underlying slow and fast dynamics of a system is instrumental for many practical applications related to the system [23, 17, 10, 22].

As a long-standing problem, learning slow and fast system dynamics has been investigated in numerous existing studies [23, 28]. However, a number of challenges still remain unsolved. First, there exist diverse time scales in different systems. Variables with the same convergence time might be the fast variables in some systems while being slow variables in other systems simultaneously. Most existing approaches simply assume the time scale to split the slow and fast variables is known [23, 28]. The first challenge is determining the appropriate time scale to separate slow and fast variables, followed by determining the dimensionality of intrinsic variables that describe slow and fast dynamics. Using excessively high-dimensional representation vectors to describe slow dynamics may mix fast dynamics into the extracted slow variables, resulting in a failure to separate them. Even when the appropriate time scale and dimensionalities are known, effectively learning slow and fast dynamics remains challenging. Currently, there is no integrated framework that can concurrently learn the slow and fast system dynamics.

In this paper, we aim to develop an integrated framework that can effectively learn slow and fast system dynamics by automatically discovering appropriate time scales to separate slow and fast variables and the dimensionalities of their intrinsic variables.

[*] Also with School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China.
[†] Huandong Wang is the corresponding author (wanghuandong@tsinghua.edu.cn).

KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599858

Specifically, we propose a novel intrinsic dimensionality (ID) driven learning method based on a time-lagged autoencoder framework, which is able to identify appropriate time scales to separate slow and fast variables and their IDs simultaneously. Further, based on the obtained time scales and IDs, we propose an integrated framework to concurrently learn the system's slow and fast dynamics. Our framework proposes a neural network based on Koopman theory to learn slow dynamics, where our prior knowledge of time scale and ID is utilized in the manner of inductive bias by constraining the Koopman operator to better extract the slow variables of the system. Furthermore, a decay rate guided autoregressive model is proposed to learn the fast dynamics, which effectively models the short-term drastic changes of fast variables and their long-term decay rate effectively guided by the obtained time scale of separating slow and fast variables as prior knowledge.

Our contribution can be summarized as follows:

- We propose a novel ID-driven learning method based on the time-lagged autoencoder framework, which is able to identify appropriate time scales to separate slow and fast variables and their IDs simultaneously.
- We propose an integrated framework to concurrently learn the slow and fast system dynamics, which utilizes our prior knowledge of time scale and ID in the manner of inductive bias and utilizes a deep Koopman neural network and a decay rate guided autoregressive model to learn the complicated slow and fast dynamics, respectively.
- Extensive experimental results show that our framework is able to efficiently learn slow and fast system dynamics. Specifically, our proposed method outperform four representative baselines, with an improvement about 36% on average in long-term prediction performance. Furthermore, our framework is able to extract interpretable slow and fast dynamics highly correlated with the known slow and fast variables in the dynamical systems.

This paper is structured as follows. We begin by introducing related works. Then, we present a mathematical model to formulate our problem and give a high-level overview of our proposed system. Next, we introduce the methodology of our proposed framework for automatically finding appropriate time scales to separate slow and fast variables and learning slow and fast system dynamics. Following our methodology, we present extensive evaluations and validations of our method. Finally, we conclude our paper.

2 RELATED WORK

Slow and fast variable detection: Singer et al. [23] present a metric to measure the distance between samples in dimensionalities corresponding to the solve variables, which utilizes the randomness of the system and needs samples of the system state to evolve for a given time period. Then, they utilize anisotropic diffusion maps and the obtained metric to extract the slow variables. Dsilva et al. [9] adopt a similar framework and propose a metric based on the Mahalanobis distance. Wehmeyer et al. [28] propose a time-lagged autoencoder for slow variable extraction, which is achieved by utilizing the encoder to extract the latent features from the current system state that can help us best predict the further system state over a certain period of time. Chen et al. [6] further utilize this framework to discover fundamental variables of complex systems. We can observe that existing methods for detecting slow and fast variables require prior knowledge of the appropriate time scale for separation of the target system. However, most of these techniques are unable to accurately determine the IDs of slow and fast variables, which substantially limits their flexibility and practical usability.

ID estimation: The ID of a system or a dataset refers to the minimum number of variables required to describe it. Due to its important role in dimensionality reduction [19], data denoising [21], outlier detection [2], and complex system comprehension [6], etc., a number of approaches have been proposed to implement ID estimation. Specifically, based on the observation that the number of data points within distance r of any given data point L_i is proportional to r^{ID} with sufficiently small r, Levina et al. [18] propose a maximum likelihood estimation (MLE) based ID estimator, which utilizes the Euclidean distance between each data point and its k nearest neighbors. Lombardi et al. [19] propose another ID estimator based on the probability density function of the normalized nearest neighbor distance named MiND. Amsaleg et al. [1] propose several estimators of local ID using well-established techniques including MLP, the method of moments (MoM), probability weighted moments (PWM), and regularly varying functions (RV). Amsaleg et al. [2] propose an ID estimator that utilizes all the pairwise distances within the given sample rather than only using the distance of neighbors. For a system with both slow and fast dynamics, both slow and fast variables contribute to its ID. Even if the fast variables converge to a pseudo-steady state in the long-term evolution, where they are slaved by the slow variables, they still dominate the initial dynamics of the system. Consequently, the existing ID estimation techniques can only estimate the sum of the degrees of freedom of fast variables and slow variables, but cannot reliably discriminate between their respective dimensionalities. Moreover, existing methods only focus on static samples, lacking an understanding of the ID of the system from the perspective of dynamic evolution. In contrast, our framework can automatically identify the appropriate time scale to separate slow and fast dynamics and utilize it as prior knowledge to learn slow and fast variables efficiently. Differently, we apply the ID estimation method from a dynamic evolution perspective, which helps us to automatically mine the IDs corresponding to fast dynamics and slow dynamics in the system.

Dynamical system modeling: With the rising paradigm of artificial intelligence, there is also a growing number of research utilizing neural networks to model dynamical systems. A number of approaches seek to directly learn the system dynamics in a data-driven manner. Zhang et al. [29] combine neural ordinary differential equations (Nerual ODE) and graph neural networks (GNN) to learn the dynamics of complex networks. Huang et al. [14] propose a latent ordinary differential equation generative model to learn the system dynamics based on irregularly-sampled observational data. Kipf et al. [16] utilize a variational autoencoder to infer the relational structure of dynamical systems. Huang et al. [15] utilize graph ordinary differential equations to jointly learn the evolution of nodes and edges of dynamical systems. Other approaches focus on modeling the system dynamics based on inductive bias, e.g., the existence of Koopman invariant subspace, symmetry, etc. Specifically, Takeishi et al. [24] propose a data-driven numerical algorithm for Koopman spectral analysis based on optimizing the residual sum of squares (RSS) loss. Azencot et al. [3] further propose a consistent Koopman autoencoder model to learn the consistent dynamics of nonlinear dynamical systems. Greydanus et al. [11] utilize neural networks to learn the system's Hamiltonian, which enables unsupervised training of models strictly satisfying conservation law. Wang et al. [27] propose a new class of approximately equivariant networks which is able to preserve symmetry in modeling dynamical systems. Different from previous studies, our work mainly focuses on modeling the dynamical system by separating and learning the slow and fast dynamics within the system. As a consequence, the utilized neural networks including the Koopman theory based neural network as well as decay rate guided autoregressive neural network are all tightly combined with the discovered time scale and IDs of slow and fast dynamics, which is reflected in their network architectures, optimization targets, etc.

3 SYSTEM MODEL AND OVERVIEW

In this section, we formulate the problem and introduce the overall system for the analysis of the slow and fast system dynamics.

3.1 Problem Formulation

We consider the evolution of dynamical systems with slow and fast dynamics, during which the slow and fast dynamical components are mixed in the observed data with nonlinear interactions. The evolution of the fast variable component is a fast process whose dynamics can only be observed on small time scales. A statistical illustration is that the autocorrelation coefficient of the fast dynamics decreases with increasing lag time. Differently, the evolution pattern of the slow dynamics can still be identified on long time scales [12]. We utilize $X(\tau) = \{x_\tau, ..., x_{n\tau} | x \in \mathbb{R}^N\}$ to denote the N-dimensional system evolving from an initial state to an observed trajectory of length n, where τ is the time scale of the observation, i.e., the time interval between adjacent sampling points.

From the perspective of slow and fast dynamics, the trajectory can be divided into a fast variable component $X_f(\tau) = \{x_{f,\tau}, ..., x_{f,n\tau}\}$ and a slow variable component $X_s(\tau) = \{x_{s,\tau}, ..., x_{s,n\tau}\}$. The dynamics of the system are embedded in the evolutionary trajectories of the slow and fast variables. Our goal is to learn the slow and fast dynamics of the system, which poses the challenge of separating the slow and fast components $X_s(\tau)$ and $X_f(\tau)$ from the original trajectory $X(\tau)$. In order to effectively identify the slow variable components and their ID D_s in the observed data, we need to analyze the change patterns of system states at different time scales and judge the dimensionality D_s of slow variables at a suitable time scale τ_s. Finally, we verify the effectiveness of the learned slow and fast dynamics of the system by predicting the future evolution state $x_{t+m\tau} (m \geq 1)$ of the observed state x_t, ranging from short-term evolution to long-term evolution.

3.2 Overall Framework

The original observations of evolutionary trajectories of dynamical systems contain information on both slow and fast dynamics, making it challenging to separate them without prior knowledge. To quantitatively analyze the system dynamics composition, we design an ID-driven time scale selection and slow variable extraction framework, and further model the slow and fast dynamics separately. Specifically, our analytical framework consists of three sub-modules.

- **ID-driven time scale selection:** The ID is the minimum dimensionality required to describe the dynamics of the system, and it reflects the number of potential variables that play a dominant role in the evolution of the system. Observing the system according to the short and long time scales τ_1 and τ_2 ($\tau_1 \ll \tau_2$) yields the evolutionary trajectories $X(\tau_1)$ and $X(\tau_2)$, respectively. Since the fast dynamics cannot be finely observed on long time scales, $X(\tau_2)$ will have a lower ID than $X(\tau_1)$. When τ_2 is large enough, $X(\tau_2)$ only responds to the dimensionality of the slow dynamics, i.e., D_s. We observe the system on a set of time scales and obtain the time scales applicable to the observation of the slow dynamics using the ID as a guide. Specifically, we obtain the system ID by analyzing the embedding obtained from a time-lagged autoencoder, which is described in detail in Sec. 4.1.
- **Secondary encoding based slow and fast separation**: On the determined time scale τ_s, we calculate the dimensionality of the slow dynamics D_s. We take D_s as the embedding dimensionality and connect the secondary encoder after the time-lagged encoder to obtain the potential representation of the slow dynamics based on the observed sequence $X(\tau)$. A decoder network is further utilized to map the representation of the slow dynamics back to the original observation space, and the slow dynamics component $X_s(\tau)$ is successfully extracted in this process to complete the fast-slow separation.
- **Learning of slow and fast dynamics**: We use Koopman theory to analyze the slow dynamics and predict changes at any continuous time by learning the linear state transfer matrix $\mathcal{K}$ of the slow variables in the Koopman subspace. For the part of the fast dynamics, we propose a decay rate guided autoregressive model to learn it, which is effectively guided by the prior knowledge of the obtained time scale of separating slow and fast variables in the form of decay rates. Finally, we couple the slow and fast dynamics to model the whole system dynamics.

4 METHOD

Learning the slow and fast system dynamics means separating their components from observed trajectories and predict their future evolution. The predictions of the slow and fast components form the system's evolutionary trajectory from the given state x_t. In this section, we first introduce the ID-driven automatic time scale selection method, which successfully selects a suitable time scale τ_s for observations without prior knowledge, and computes the ID D_s of the slow variables based on the observed sequence $X(\tau_s)$ at the time scale τ_s. We then introduce an integrated learning framework for slow and fast dynamics, which performs a secondary encoding and decoding of the original observed trajectory $X(\tau)$ to obtain the slow dynamic component $X_s(\tau)$ as well as the fast dynamic component $X_f(\tau)$, completing the slow and fast separation. Finally, a Koopman theory based neural network and a prior knowledge guided recurrent neural network are utilized to learn the slow and fast dynamics of the system, respectively.

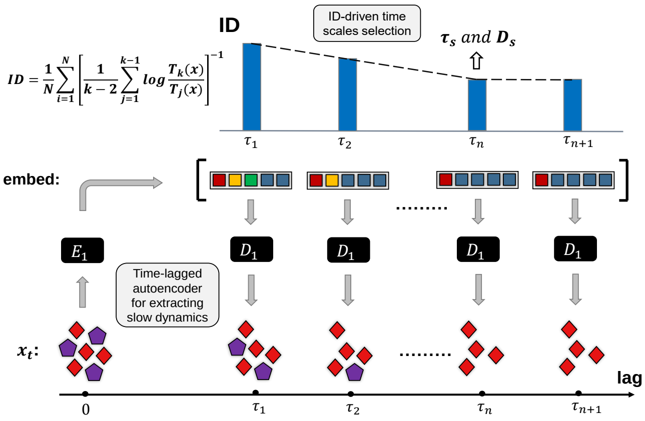

ID
ID-driven time scales selection
τ_s and D_s
$ID = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{k-2}\sum_{j=1}^{k-1}\log\frac{T_k(x)}{T_j(x)}\right]^{-1}$
τ_1
τ_2
τ_n
τ_{n+1}
embed:
E_1
D_1
Time-lagged autoencoder for extracting slow dynamics
x_t:
lag
0
τ_1
τ_2
τ_n
τ_{n+1}

Figure 1: An illustration of the ID-driven time scale selection module

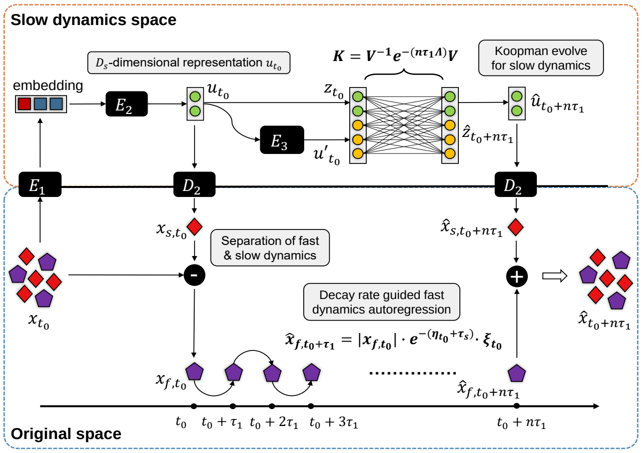

Slow dynamics space
D_s-dimensional representation u_{t_0}
$K = V^{-1}e^{-(n\tau_1\Lambda)}V$
Koopman evolve for slow dynamics
embedding
E_2
u_{t_0}
z_{t_0}
$\hat{u}_{t_0+n\tau_1}$
E_3
u'_{t_0}
$\hat{z}_{t_0+n\tau_1}$
E_1
D_2
D_2
x_{s,t_0}
Separation of fast & slow dynamics
$\hat{x}_{s,t_0+n\tau_1}$
x_{t_0}
Decay rate guided fast dynamics autoregression
$\hat{x}_{t_0+n\tau_1}$
$\hat{x}_{f,t_0+\tau_1} = |x_{f,t_0}| \cdot e^{-(\eta_{t_0}+\tau_s)} \cdot \xi_{t_0}$
x_{f,t_0}
$\hat{x}_{f,t_0+n\tau_1}$
t_0
$t_0+\tau_1$
$t_0+2\tau_1$
$t_0+3\tau_1$
$t_0+n\tau_1$
Original space

Figure 2: Separation and evolvement model of the slow and fast dynamics

4.1 ID-Driven Selection of Time Scales

In order to provide a quantitative description of the dynamics of the system at different time scales τ, we measure the ID of the system by geometric manifold learning algorithms. The variation of ID with τ reveals the complicated interactions between slow and fast dynamics of the system at different time scales, which is a key guide to separate them. However, existing ID analysis methods are unable to compute the dimensionality of fast or slow variables directly from the trajectories obtained from the system with mixed slow and fast dynamics. Therefore, our goal is to design an automatic analysis mechanism that can calculate the dimensionality of the slow and fast variables in the observed data of the system.

For a given N-dimensional system, we design a time-lagged autoencoder with lag τ to compute the M-dimensional embedding of the system state, and obtain the ID at the time scale τ by computing the geometric dimensionality of the obtained embedding. The time-lagged autoencoder consists of an encoder $E_1 : \mathbb{R}^N \to \mathbb{R}^M$ and a decoder $D_1 : \mathbb{R}^M \to \mathbb{R}^N$. The design of this module is shown in Figure 1. By learning the prediction task with lag τ, the time-lagged autoencoder compresses the dynamics of the system at the time scale τ into the embedding vector in the latent space. We design the loss function for the time-lagged prediction task as

$$L_1 = \|x_{t+\tau} - D_1(E_1(x_t))\|. \tag{1}$$

As network parameters are updated, the encoder E_1 learns the ability to compress redundant information from the raw observations and extract relatively slower dynamics with respect to the lag τ.

The number of variables that dominate the system dynamics can be obtained by calculating the ID of the embedding through the geometric manifold learning algorithm. For a vector set $\{L_1, L_2, ..., L_N\}$, the number of data points within distance r of any given data point L_i is proportional to r^{ID}, when r is small [6]. This leads to the local ID near L_i as $\left[\frac{1}{k-2}\sum_{j=1}^{k-1}\log\frac{T_k(x)}{T_j(x)}\right]^{-1}$, where $T_k(x)$ is the Euclidean distance between L_i and its k nearest neighbors. We divide by $k-2$ rather than $k-1$ to make the estimator asymptotically unbiased [18]. The global ID estimate of the embedding is obtained by taking the average as

$$ID = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{k-2}\sum_{j=1}^{k-1}\log\frac{T_k(x)}{T_j(x)}\right]^{-1}. \tag{2}$$

We denote the ID of the embedding as D_τ when lag is τ. As τ increases, the fast dynamical information evolving on the tiny scale will not be observed and the ID shows a decreasing trend. The observed time scale τ_s suitable for separating slow and fast dynamics should satisfy: no fast dynamics are observed at all in $X(\tau_s)$, while the system ID is equal to the slow variable dimension, i.e., $D_{\tau_s} = D_s$. As τ increases, starting from τ_s, the measured IDs fully reflect the dimensionality of the slow variable and no longer change. Starting from the minimum observation interval τ_1 of the system, given a sequence of time scales $T = (\tau_1, \tau_2, \ldots, \tau_n, \ldots, \tau_N)$, we define the appropriate time scales for the slow variables as

$$\tau_s = \min\left(\{\tau_n \mid \left|D_{\tau_n} - D_{\tau_m}\right| < 1,\ n < \forall m \le N\}\right). \tag{3}$$

Note that the ID of the slow dynamics D_s is determined by its own properties and does not change with the observation scale τ. Therefore, D_s computed at time scale τ_s is also applicable to other time scales τ. Similarly, the time-lagged encoder E_1 trained with lag τ_s is able to extract slow variable information from the observed trajectory at arbitrary time scale τ, about which we will perform a detailed validation in the experimental section.

4.2 Koopman Modeling of Slow Dynamics

In Koopman operator theory, a finite-dimensional nonlinear dynamical system is mapped into phase space by an observation function g, and its evolution along the phase space orbit is described by an infinite dimensional linear operator $\mathcal{K}$. For the dynamical system $x_{t+dt} = f(x_t), x \in \mathbb{R}^d$, Koopman analysis requires solving the approximation of the Koopman operator $\mathcal{K}$ and the observation function g to satisfy $\mathcal{K}g(x) = g(f(x))$ [20].

In this module, we use the obtained ID of the slow dynamics D_s and the time-lagged encoder trained at the time scale τ_s to encode the original observations by the secondary encoder E_2 to obtain the representation u of the slow dynamics of the system. Considering that the Koopman operator theory boosts the observation dimensionality to map low-dimensional nonlinear dynamics to linear high-dimensional spatial evolution [5], we boost u to complete the mapping to the K-dimensional Koopman subspace. In the Koopman subspace, the slow dynamics evolve in a linear fashion and are characterized by the Koopman operator $\mathcal{K}$. It is subsequently mapped back to the original space by the decoder D_2. The whole process is shown in Figure 2.

4.2.1 ***Koopman invariant subspace of slow variable.*** In Sec. 4.1, we have obtained the time scale τ_s for observing the slow dynamics and the corresponding time-lagged autoencoder, based on which we are able to extract the M-dimensional representation of the slow variables from the original trajectory $X(\tau)$. Subsequently, based on the identified slow dynamics dimensionality D_s, a second level encoder $E_2 : \mathbb{R}^M \to \mathbb{R}^{D_s}$ is designed to obtain the slow dynamics representation $u = E_2(E_1(x))$. We pass u through an encoder E_3 to obtain the nonlinear combinatorial terms u', and concatenate these combinatorial terms with u as the K-dimensional Koopman variable $z = \{u, u'\} \in \mathbb{R}^K$. In turn, we intercept the previous D_s dimensionality of z to get u. During training, E_1 will be frozen to maintain its ability to extract the slow dynamics. Finally, we design a decoder $D_2 : \mathbb{R}^{D_s} \to \mathbb{R}^N$ to map the representation u of the slow dynamics back to the original space and obtain the slow dynamic component of the observed sequence as follows:

$$X_s(\tau) = D_2(E(X(\tau))) = D_2(E_2(E_1(X(\tau)))), \tag{4}$$

where $E(\cdot) = E_1(E_2(\cdot))$. At this point, the components of fast dynamics can be obtained by subtracting the components of slow dynamics from the overall dynamics of the system as follows:

$$X_f(\tau) = X(\tau) - X_s(\tau). \tag{5}$$

The component of slow dynamics of the original data is not available due to the absence of prior knowledge. This poses a challenge to the supervised training of E_2, E_3, and D_2. We designed a multi-task loss function

$$L_2 = \|X(\tau) - D_2(E(X(\tau)))\| + \alpha |E_1(X(\tau)) - E_1(D_2(E(X(\tau))))| \tag{6}$$

to approximate the goal of slow variable extraction. The first term of (6) represents the reconstruction error, which guarantees the accuracy of the mapping of D_2 from the slow dynamics subspace to the original space. However, this error alone can mislead E_2 to compress the components of fast dynamics into the representation of slow dynamics, which helps the reconstruction task but interferes with the extraction of slow variables. Therefore, we designed the second term as a penalty, which constrains E_2 and D_2 to compress and recover only the slow dynamic components by the ability of E_1 to identify the slow dynamics. α is the penalty coefficient.

4.2.2 ***Learning-based Koopman operator of slow dynamic.*** The Koopman operator $\mathcal{K}$ linearly characterizes the dynamics of the system in the Koopman subspace. We parameterize $\mathcal{K}$ with the following eigen decomposition structure,

$$\mathcal{K}(\tau) = V^{-1} \exp(\tau\Lambda) V, \tag{7}$$

where $exp(\tau\Lambda) = diag(exp(\tau\lambda_1), ..., exp(\tau\lambda_K)) \in C^{K\times K}$ is the diagonal matrix of eigenvalues, $exp(\tau\lambda_k)$ is the kth eigenvalue, and $V = [v_1, .., v_K] \in C^{K\times K}$ is the set of eigenvectors and v_k is the kth eigenvector. The Koopman state transfer of slow dynamics in the time step τ can be described by $\mathcal{K}(\tau)$ after determining V and Λ,

$$\hat{z}_{t+\tau} = \mathcal{K}(\tau) z_t = V^{-1} \exp(\tau\Lambda) V z_t. \tag{8}$$

Specifically, V and Λ are learned through performing gradient descent updates by back-propagating the error of the prediction task. The error consists of two specific parts, and the first is the evolution error in both slow representation space and Koopman space:

$$\begin{aligned} L_3 &= \|u_{t+\tau} - \hat{u}_{t+\tau}\| + \|z_{t+\tau} - \hat{z}_{t+\tau}\| \\ &= \|u_{t+\tau} - \hat{u}_{t+\tau}\| + \|z_{t+\tau} - \mathcal{K}(\tau) z_t\|, \end{aligned} \tag{9}$$

where $u_t = E(x_t)$ and $z_t = \{u_t, E_3(u_t)\}$. The other part is jointly calculated by the slow and fast variable prediction results in (11), which will be introduced in detail in the following section.

4.3 Decay Rate Guided Autoregression for Fast Dynamics

In Sec. 4.2, we have extracted the component of fast dynamics $X_f(\tau)$ from $X(\tau)$ and now we model its evolution mechanism. In our ID-driven analysis in Sec. 4.1, we have selected a time scale τ_s suitable for separating the slow and fast dynamics, beyond which the fast dynamics will not be observed. Under the guidance of τ_s, we model the decay process of fast dynamics as

$$\hat{x}_{f,t+\tau_1} = |x_{f,t}| \cdot \exp(-(\eta_t + \tau_s)) \cdot \xi_t, \tag{10}$$

where ξ_t denotes the direction of the evolution of the fast dynamics at moment t and η_t denotes the relative decay rate of the fast dynamics at this time, which is constrained to be greater than 0. η_t and ξ_t are determined by both the state of the slow and fast dynamics at moment t. We use a recurrent neural network, i.e., the long short-term memory (LSTM) [13] , to implement their predictions. The inputs of the LSTM are the slow and fast dynamic components $x_{s,t}$, $x_{f,t}$ at moment t, and the outputs are η_t and ξ_t. Using the minimum observation interval τ_1 of the original sequence as the unit time step for the prediction of the fast dynamics, the long-time prediction is accomplished in an autoregressive manner. For simplicity, we denote the evolution mechanism modeled by (10) as $\hat{x}_{f,t+\tau_1} = f(x_{f,t}, x_{s,t})$.

The prediction of the whole system dynamics is equal to the respective prediction of the slow and fast dynamics, i.e. $\hat{x}_{t+\tau} = \hat{x}_{s,t+\tau} + \hat{x}_{f,t+\tau}$. The loss function is defined as

$$\begin{aligned} L_4 &= \|x_{t+\tau} - \hat{x}_{t+\tau}\| \\ &= \|x_{t+\tau} - \hat{x}_{s,t+\tau} - \hat{x}_{f,t+\tau}\| \\ &= \|x_{t+\tau} - V^{-1}\exp(\tau\Lambda) V E(x_t) - \underbrace{f \circ ... \circ f}_{n\ times}(x_t - D_2(E(x_t)), E(x_t))\|, \end{aligned} \tag{11}$$

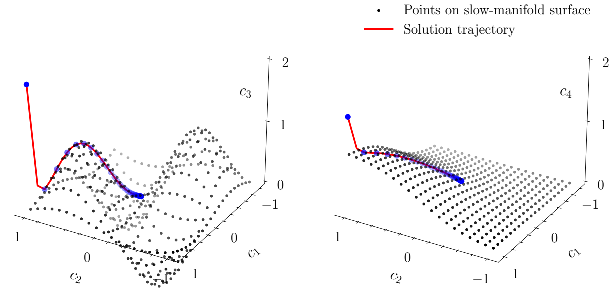
where the fast dynamics undergo a total of $n = \frac{\tau}{\tau_1}$ evolutions in an autoregressive manner.

4.4 Training

The training of the whole framework is divided into two stages. First, we follow the given time-scale series $T = (\tau_1, \tau_2, ..., \tau_n, ..., \tau_N)$ to train the time-lagged autoencoders E_1 and D_1 with different lag values, respectively, and the loss function is L_1. From the variation of the dimensionality of the embedding with different lag values, the time scale τ_s suitable for separating the slow dynamics is selected and the dimensionality D_s corresponding to the slow dynamics is recorded. The learned model parameters of E_1 are reserved for the second stage of training. Next, we train the overall framework for fast-slow separation and dynamics modeling end-to-end, which consists of: a secondary encoder E_2, a nonlinear generator E_3, a decoder D_2, an LSTM, eigenvalues Λ, and eigenmatrix V of the Koopman operator $\mathcal{K}$. The LSTM performs a $\frac{\tau}{\tau_1}$-step autoregressive prediction of the fast dynamics with τ_1 as the single-step time interval, and the slow dynamics are also trained for the prediction with the same time interval. In the pre-training period, we freeze E_1 and guide the model through multi-task learning with the loss function $L = L_2 + L_3 + L_4$, where L_3 and L_4 are averaged over the $\frac{\tau}{\tau_1}$-step predictions. After training enough rounds, we turn down the learning rate and liberate the E_1 weight parameters to involve it in the end-to-end fine-tune training.

5 EXPERIMENTS

In this section, we apply the proposed framework to capture the slow and fast dynamics of two dynamic systems and predict their evolutionary behavior. We first introduce the dynamics of the systems to be studied, then compare the accuracy of our model with other baseline models in the prediction task. Finally, we analytically validate the effectiveness of the proposed framework for slow and fast dynamics separation and modeling.



(a) Evolution of c_3

(b) Evolution of c_4

Figure 3: Fast-variable evolutionary trajectory of the 4-dimensional system.

5.1 Experimental settings

System Dynamics. We test the proposed model in two slow and fast systems investigated by existing studies. The analysis of the dynamics of the two systems is presented below.

- **1S2F System:** This system contains one slow variable and two fast variables (1S2F) and is an inherently nonlinear chemical reaction system [23] in a black box which is charged into two molecules X and Y from the outside at a specified rate. The chemical reactions with different rate coefficients occurring inside the black box are:

$$X \xrightarrow{k_1} X + Z, \quad Y + Z \xrightarrow{k_2} Y \tag{12}$$

$$\emptyset \xrightarrow{k_3} Y, \quad Y \xrightarrow{k_4} \emptyset \tag{13}$$

$$\emptyset \xrightarrow{k_5} X \tag{14}$$

We select the value of the rate constant as

$$k_1 = 1000, \quad k_2 = 1, \quad k_3 = 40, \quad k_4 = 1, \quad k_5 = 1, \tag{15}$$

which ensures that the reaction in (12) is the fastest, the reaction in (13) occurs on an intermediate time scale, and the reaction in (14) is the slowest. The system has three observed variables X, Y, and Z, which represent the quantities of the three reactants, respectively. The model of this reaction can be approximated by the ODE system with variables $x = \frac{X}{100}$, $y = \frac{Y}{40}$ and $z = \frac{Z}{2500}$ as

$$\begin{cases} \dfrac{dx}{dt} = \dfrac{k_5}{100}, \\ \dfrac{dy}{dt} = \dfrac{k_3}{40} - k_4 y, \\ \dfrac{dz}{dt} = \dfrac{100k_1 x}{2500} - 40k_2 yz. \end{cases} \tag{16}$$

The equation shows that X increases gradually at a slow rate, the equilibrium-state value of Y is 40, and the equilibrium-state approximation of Z is $Z = X/Y$. Due to the stochastic nature of the chemical reaction, Y will continue to fluctuate after a rapid convergence to the equilibrium state. Consequently, the fluctuations of Y lead to the dynamics of Z, and it is X and Y that determine the evolution of the system. The evolution of this system can be fully described by the slow dynamics of the long-term growth of X and the high-frequency noise term of Y. The goal of our model is to identify the true slow dynamic term X and make accurate predictions of the long-term evolution of the system through learning the dynamics of X.

- **2S2F System:** This system is a 4-dimensional ODE system [8] which contains two slow variables and two fast variables(2S2F) with the dynamics equations:

$$\begin{cases} \dfrac{dc_1}{dt} = f_1(c_1, c_2), \\ \dfrac{dc_2}{dt} = f_2(c_1, c_2), \\ \dfrac{dc_3}{dt} = -\dfrac{1}{\epsilon}[c_3 - \theta_1(c_1, c_2)] + f_1 \partial c_1 \theta_1(c_1, c_2) + f_2 \partial c_2 \theta_1(c_1, c_2), \\ \dfrac{dc_4}{dt} = -\dfrac{1}{\epsilon}[c_4 - \theta_2(c_1, c_2)] + f_1 \partial c_1 \theta_2(c_1, c_2) + f_2 \partial c_2 \theta_2(c_1, c_2), \end{cases} \tag{17}$$

where ϵ and ∂i denote fixed small quantities and bias derivatives with respect to variable i, respectively. The dynamics of the two fast variables (c_3 and c_4) are governed by the motion of the slow variables c_1 and c_2, and the functions f_1, f_2, θ_1 and θ_2 depend only on c_1 and c_2. The Chapman-Enskog solution of the equation shows that c_3 and c_4 will converge to θ_1 and θ_2, respectively, after stabilization. In the following experiment, f_1, f_2, θ_1 and θ_2 are selected as

$$\begin{aligned} f_1(c_1, c_2) = -c_1, \quad f_2(c_1, c_2) = -2c_2, \\ \theta_1(c_1, c_2) = sin(\omega c_1) sin(\omega c_2), \\ \theta_2(c_1, c_2) = [(1 + e^{-\omega c_1})(1 + e^{-\omega c_2})]^{-1}, \end{aligned} \tag{18}$$

which means that c_1 and c_2 will eventually converge to 0, and c_2 is faster than c_1. Before c_1 and c_2 converge, c_3 and c_4 will quickly converge to the surfaces inscribed by θ_1 and θ_2 (see Figure 3), respectively, and then move slowly with c_1 and c_2.

In this system, the goal of our model is to identify the slow and fast components and learn their evolutionary properties before the dynamics dissipate as the system converges.

Datasets and Baselines. In the 1S2F system, we take the system from a random initial state and simulate the evolution of the system in 15.1s using the Gillespie stochastic simulation algorithm. The result of a single simulation is shown in Figure 4. Subsequently, we sample the simulation sequence twice at the intervals of 10^{-2}s to ensure that the observation sample points are equally spaced. The above operation is repeated to obtain 100 observed trajectories of the evolution process to obtain the dataset. In the second experiment, we randomly initialize 4 variables from (-3, 3) to numerically solve the evolution trajectory of the differential equation in 5.1s with a time unit of 10^{-2}s. We evolve 200 observed trajectories with different random seeds to construct the dataset. The training, validation, and test sets are divided in the ratio 7:1:2 in both experiments.

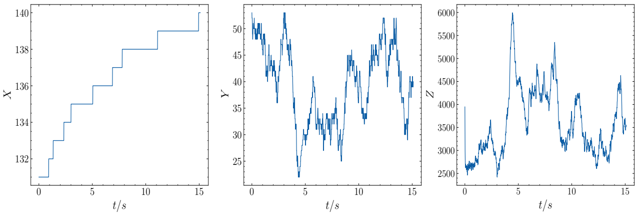


Figure 4: Evolutionary process of X, Y, and Z of a single stochastic simulation trajectory

We choose the following models as baselines in the long-time prediction task:

- **LSTM** [13] is an RNN-based sequence-to-sequence prediction model. It improves the long-term dependency problem of simple RNN and is suitable for processing and predicting long-time sequences.
- **TCN** [4] is a prediction model that captures time series information using a convolutional network. Convolutional networks avoid the gradient explosion or vanishing problem common to recursive models and perform better than RNNs in many time-series prediction tasks.
- **Informer** [30] improves the operational efficiency of the Transformer model. It proposes the ProbSpare self-attention, self-attention distillation, and parallel generative decoder mechanism, which is getting much attention in long-time sequence prediction tasks. The amount of data required, however, is particularly high because of its massive number of trainable parameters.
- **Neural ODE** [7] is a deep learning operator for solving differential equations. It can be used in any framework to achieve temporal prediction by solving differential equations for future sequences with respect to the input sequence. Neural ODE enables prediction for any continuous time by integrating over continuous time with respect to the differential equations.

In following experiments, all models are fully trained, where LSTM, TCN, and Neural ODE are implemented by calling APIs provided by frameworks such as pytorch, and Informer is applied through the open-source code provided by the author. We use the default hyperparameters suggested by the authors when applying Informer.

5.2 Overall performance

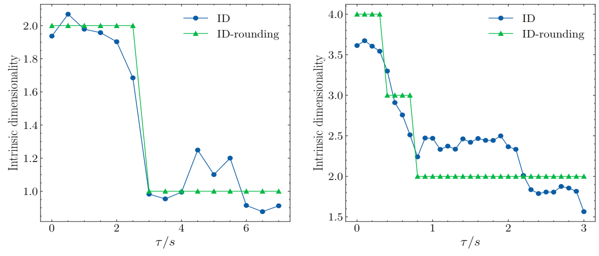
We compare our proposed model for learning slow and fast system dynamics with baseline algorithms for multivariate time series prediction at different prediction lengths. Accurate long-term predictions demonstrate the effectiveness of our model in learning the system dynamics.

In order to verify that the models really learn the dynamics of the system, the model is fed with only one observation x_t of the system and performs multi-step prediction in the autoregressive manner. Our model, LSTM, TCN, and Neural ODE are set to have a maximum prediction step of $10\tau_1$ in training to ensure that they learn the dynamics of the system only through observations within a limited view and are required to make long-term predictions (up to $50\tau_1$) in testing. Table 1 shows the comparison of the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) of the models at three extrapolation duration τ, where τ is $\{1, 10, 50\}$ times as much as the minimum observation interval τ_1.

In the comparison of results, Informer is not suitable for such limited observation learning tasks and has the worst performance due to the huge amount of data required. Other models perform similarly on short time scales, but our model that truly learns the slow and fast dynamics achieves the best performance as the required extrapolation time increases.

5.3 Analysis of slow and fast dynamics learning

To verify the effectiveness of the proposed framework for the separation of slow and fast variables, the following analysis shows how the model determines the appropriate time scale τ_s and the slow variable dimensionality d_s in two scenarios, and compares the relationship between the extracted slow variables under Koopman space and the actual slow variables of the system.



(a) 1S2F system

(b) 2S2F system

Figure 5: ID-τ curves, quantitatively characterize the dynamics at different time scales. The ID-rounding term is the result of rounding.

5.3.1 ***Determination of time scale and slow dynamics dimensionality.*** The time scale τ_s applicable to the separation of slow dynamics is first selected by our ID-driven approach. In the experiments, the minimum observation interval τ_1 is set as 0.1 and 0.3 for the two systems, respectively. The time-lagged autoencoder learns the observable trajectories of the systems with integer multiples of τ_1 as the lag values. The geometric dimensionality of embedding with lag parameter τ is plotted in Figure 5. According to the

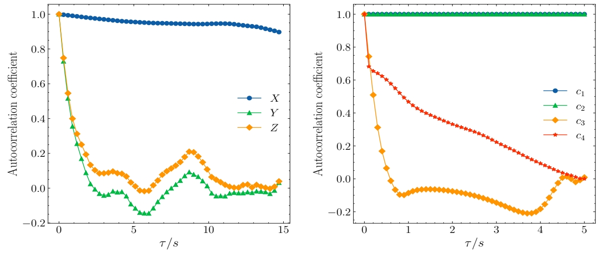
Table 1: Average performance comparison in 10 random seeds with input length 1 and prediction length $\{1, 10, 50\}$. A lower RMSE and MAPE indicate better performance. The best results are highlighted in bold.

	1S2F						2S2F					
	$\tau_1 = 0.3s$		$\tau_{10} = 3.0s$		$\tau_{50} = 15.0s$		$\tau_1 = 0.1s$		$\tau_{10} = 1.0s$		$\tau_{50} = 5.0s$	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
TCN	326.72	13.76%	381.11	16.49%	529.37	28.60%	0.1352	1.68%	0.1055	1.81%	0.0969	1.89%
LSTM	211.45	6.55%	323.61	10.93%	334.74	14.42%	0.0656	0.53%	0.0284	0.71%	0.0253	0.75%
Neural ODE	224.14	6.86%	334.23	10.47%	331.03	12.09%	0.0802	0.47%	0.0636	0.67%	0.0174	0.56%
Informer	742.34	13.58%	878.96	16.92%	803.55	16.64%	0.4109	18.88%	0.7146	4.58%	1.2356	3.12%
Our	221.76	6.95%	305.99	10.41%	309.51	11.74%	0.0638	0.51%	0.0242	0.49%	0.059	0.16%
Percentages	**-4.88%**	**-6.11%**	**5.44%**	**0.57%**	**6.50%**	**2.89%**	**2.74%**	**-6.3%**	**14.79%**	**26.87%**	**66.09%**	**71.43%**

determination condition of τ_s, the appropriate observation time scales for the slow variables of the two systems are 3.0s and 0.8s, respectively. The variation curves of ID in Figure 5(a) show that the chemical 1S2F system has 2 degrees of freedom at small time scales, and they actually represent the slowly increasing dynamic term of X and the noise term of Y, respectively. As τ increases, the autocorrelation of the noise term weakens to the point where it cannot be learned (see Figure 6) so that the measured ID contains only the slow dynamic term of X. Similarly, when τ is greater than 0.8, only two dimensionalities of slow variable are retained in the 2S2F system. Our model successfully identifies two systems with 1-dimensional and 2-dimensional slow variables, respectively, which coincide with the facts.

5.3.2 *Separate of slow and fast dynamics.*

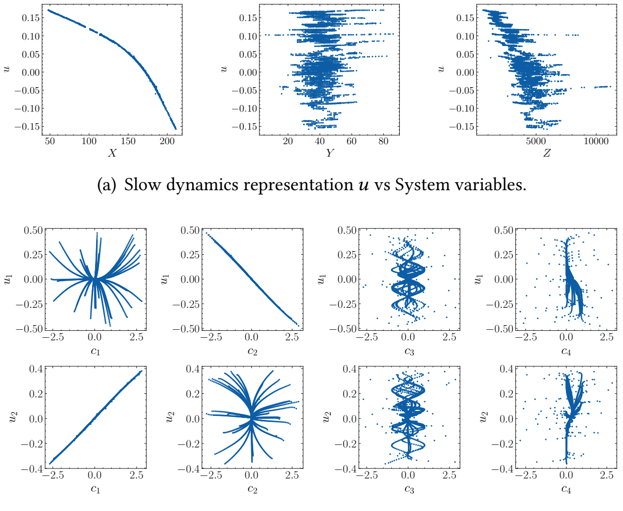
According to the conclusion of the previous step, the dimensionality of the representations vector u of the slow dynamics of the two systems are set to 1 and 2, respectively, and the second-level encoder E_2 is trained to obtain the representations u. The obtained representations u are shown against the original variables in Figure 7. In the 1S2F system, the one-dimensional slow dynamics representation u has a significant correspondence with the system variable X. In the 2S2F system, the dimensionality of representation u corresponds to the real slow variables c_1 and c_2 of the system, respectively. These results validate the effectiveness of the proposed framework for slow dynamics extraction.



(a) 1S2F system (b) 2S2F system

Figure 6: Autocorrelation coefficient of system variables.

To further illustrate the slow dynamics component of the system in the observation space, we map the extracted slow dynamics representation back to the observation space, whose results are shown in Figure 8. We can observe that variable X of the 1S2F system is fully identified as slow dynamics, however, only the steady-state value 40 of variable Y is identified as its slow dynamics. This result is encouraging, because it indicates that our model identifies the true evolutionary properties of the system on long time scales, while noise factors that are not significant for long-time evolution are stripped out. In the experiments of the 2S2F system, c_1 and c_2 are successfully identified, and it is noticed that part of the evolutionary process of c_3 and c_4 is also identified as slow dynamics. The explanation of this phenomenon is that the evolutionary process of c_3 and c_4 contains 3 stages: (1) rapid convergence from the initial state to a slow manifold governed by c_1 and c_2. This process occurs on a tiny time scale, and therefore c_1 and c_2 have not yet converged at this point; (2) subsequently, c_3 and c_4 are governed by c_1 and c_2 and evolve at the same rate in equilibrium, which exhibits the slow dynamics property; (3) finally, c_1 and c_2 generally converge to the steady state and are not changing, and thus c_3 and c_4 also remain constant. In stage (1), c_3 and c_4 exhibit fast dynamics characteristics and are sieved out by the model, while stages (2) and (3) are captured as slow dynamics components. The above results again confirm that our model has the ability to identify the slow dynamics hidden in observed data, and this accurate identification provides valuable insights for understanding complex dynamic systems.



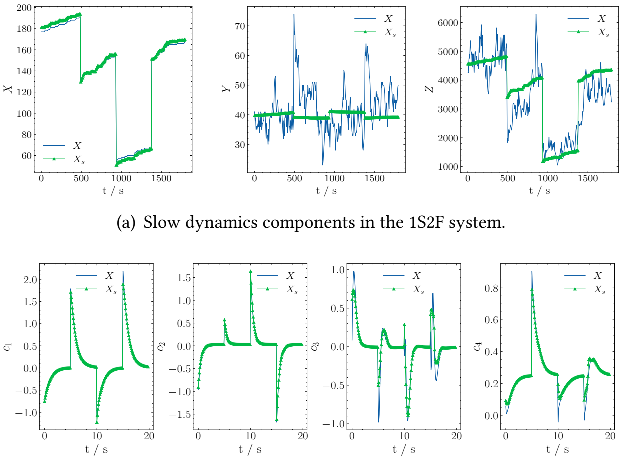
(a) Slow dynamics representation u vs System variables.

(b) Slow dynamics representation u_1, u_2 vs System variables.

Figure 7: Scatter of representation and observed variables.

5.3.3 *Long and short prediction performance analysis.*

Learning the slow dynamics accurately is crucial for the effectiveness in long-term prediction tasks, as it dominates the system's long-term evolution. The linear evolutionary nature of the system in the Koopman subspace makes our model work well in learning the slow dynamics. In fact, when the V and Λ of the constituent operators $\mathcal{K}$ are successfully fitted, the values of the slow dynamics components at any successive moments can be accurately computed, which is fundamentally different from other models.

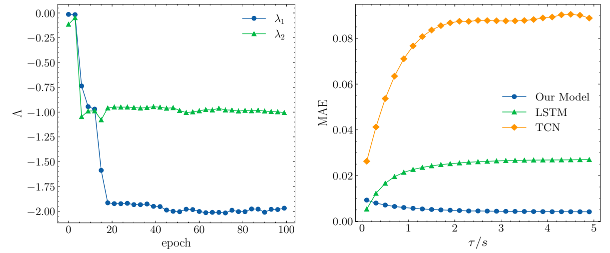


(a) Slow dynamics components in the 1S2F system.

(b) Slow dynamics components in the 2S2F system.

Figure 8: Slow dynamic components of two systems, as exemplified by the 4-segment evolutionary trajectory.

Figure 9(a) shows the learning curves of the two eigenvalues λ_1 and λ_2 of the operator $\mathcal{K}$ for the component of the 2-dimensional slow dynamics in the experiment of the 2S2F system. Corresponding to that in Figure 7(b), E_2 maps the slow dynamic components into the slow dynamics subspace with the representation components u_1 and u_2 matching the slow variables c_2 and c_1, respectively. λ_1 and λ_2 eventually converge to around -2 and -1, which represent the evolution rate coefficients of the slow representations u_1 and u_2, respectively. This result is consistent with the coefficients of the first two terms in (18). We compare the average Mean Absolute Error (MAE) changes of LSTM, TCN, and our model for c_1 and c_2 in long-time prediction, thus testing the effectiveness of the proposed model in the slow dynamics long-time evolution prediction problem. The error comparison curves are shown in Figure 9(b), where our error does not grow catastrophically due to the longer prediction time, which proves the superiority of our model for slow dynamics learning.

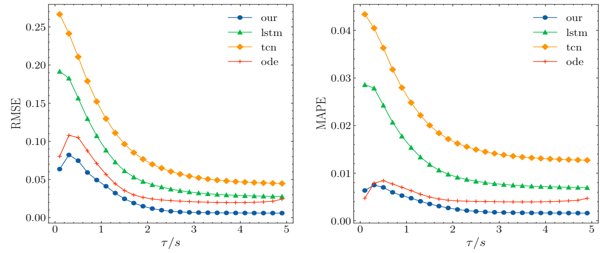


(a) Fitting process of Λ.

(b) Average MAE of the long-term prediction of c_1,c_2 in system2.

Figure. 9: Slow dynamics learning performance in our model.

Short-term predictions rely heavily on the fast dynamics, as the slow dynamics evolve slowly and are easy to predict. Therefore, the prediction performance of fast dynamics becomes an important factor in overall performance. Figure 10 illustrates the performance of multi-step predictions based on a single input sample in the 2S2F system as a function of the single-step time scale τ. Obviously, predicting fast variables is more challenging, but our model gradually achieves the best performance.



(a) RMSE

(b) MAPE

Figure. 10: Overall prediction performance for 2S2F system.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we explore the dynamics of slow and fast systems by using ID-driven learning based on time-lagged autoencoders to discover appropriate time scales and slow dynamics dimensionalities. We successfully separate the slow and fast dynamics components of observed trajectories of the system and analyze the slow dynamics using Koopman theory and the fast dynamics through the decay rate-guided autoregression. By combining these methods, we can cooperatively learn the coupled slow and fast variables. Finally, we test the proposed framework in two representative systems for time scale selection, slow and fast variable separation, and long-term prediction performance. The results show that our framework can effectively learn slow and fast system dynamics.

The success of our proposed framework on two representative systems demonstrated in Sec. 5 indicates its readiness for similar real-world scenarios. Experiment results also show that our method is feasible for online services in terms of robustness (See Appendix for details). In future work, we plan to deploy the framework as an online scientific computing service for disease propagation [25], chemical reactions [23], and neuronal potential changes [26], etc. Specifically, users can upload data and receive the results in terms of the time scale and dimensionalities of slow and fast variables. In addition, we plan to explore the fast and slow dynamics of complex networks with more complex topologies, and consider the modeling approach of Neural ODE with the advanced ability of continuous time prediction.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under 2020AAA0106000, the National Natural Science Foundation of China under U22B2057, 62272260, 62171260, 92270114, the Guoqiang Institute, Tsinghua University under 2021GQG1005 and the Young Elite Scientists Sponsorship Program by CIC (Grant No.2021QNRC001).

REFERENCES

[1] Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. 2015. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 29–38.

[2] Laurent Amsaleg, Oussama Chelly, Michael E Houle, Ken-Ichi Kawarabayashi, Miloš Radovanović, and Weeris Treeratanajaru. 2019. Intrinsic dimensionality estimation within tight localities. In *Proceedings of the 2019 SIAM international conference on data mining*. 181–189.

[3] Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. 2020. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*. 475–485.

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

[5] Steven L Brunton, Marko Budisic, Eurika Kaiser, and J Nathan Kutz. 2022. Modern Koopman Theory for Dynamical Systems. *SIAM Rev.* 64, 2 (2022), 229–340.

[6] Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. 2022. Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science* 2, 7 (2022), 433–442.

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

[8] Eliodoro Chiavazzo. 2012. Approximation of slow and fast dynamics in multiscale dynamical systems by the linearized Relaxation Redistribution Method. *J. Comput. Phys.* 231, 4 (2012), 1751–1765.

[9] Carmeline J Dsilva, Ronen Talmon, C William Gear, Ronald R Coifman, and Ioannis G Kevrekidis. 2016. Data-driven reduction for a class of multiscale fast-slow stochastic dynamical systems. *SIAM Journal on Applied Dynamical Systems* 15, 3 (2016), 1327–1351.

[10] Andrew L Ferguson, Athanassios Z Panagiotopoulos, Pablo G Debenedetti, and Ioannis G Kevrekidis. 2010. Systematic determination of order parameters for chain dynamics using diffusion maps. *Proceedings of the National Academy of Sciences (PNAS)* 107, 31 (2010), 13597–13602.

[11] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. 2019. Hamiltonian neural networks. *Advances in neural information processing systems* 32 (2019).

[12] Peter R Hansen and Asger Lunde. 2014. Estimating the persistence and the autocorrelation function of a time series that is measured with error. *Econometric Theory* 30, 1 (2014), 60–93.

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

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

[15] Zijie Huang, Yizhou Sun, and Wei Wang. 2021. Coupled graph ode for learning interacting system dynamics. In *The 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[16] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. 2018. Neural relational inference for interacting systems. In *International conference on machine learning*. 2688–2697.

[17] Chang Hyeong Lee and Hans G Othmer. 2010. A multi-time-scale analysis of chemical reaction networks: I. Deterministic systems. *Journal of mathematical biology* 60, 3 (2010), 387–450.

[18] Elizaveta Levina and Peter Bickel. 2004. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems* 17 (2004).

[19] Gabriele Lombardi, Alessandro Rozza, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. 2011. Minimum neighbor distance estimators of intrinsic dimension. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*. 374–389.

[20] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications* 9, 1 (2018), 4950.

[21] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*. 3355–3364.

[22] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. 2018. VAMPnets for deep learning of molecular kinetics. *Nature communications* 9, 1 (2018), 5.

[23] Amit Singer, Radek Erban, Ioannis G Kevrekidis, and Ronald R Coifman. 2009. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. *Proceedings of the National Academy of Sciences (PNAS)* 106, 38 (2009), 16090–16095.

[24] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. 2017. Learning Koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems* 30 (2017).

[25] Ying Tang, Jiayu Weng, and Pan Zhang. 2023. Neural-network solutions to stochastic reaction networks. *Nature Machine Intelligence* (2023), 1–10.

[26] Pantelis R Vlachas, Georgios Arampatzis, Caroline Uhler, and Petros Koumoutsakos. 2022. Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence* 4, 4 (2022), 359–366.

[27] Rui Wang, Robin Walters, and Rose Yu. 2022. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*. 23078–23091.

[28] Christoph Wehmeyer and Frank Noé. 2018. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of chemical physics* 148, 24 (2018), 241703.

[29] Chengxi Zang and Fei Wang. 2020. Neural dynamics on complex networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 892–902.

[30] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.

A APPENDIX

A.1 Robustness Analysis

We test whether the framework can work robustly in the 2S2F system with important hyperparameters, including different slow variable dimension D_s, Koopman dimension K, and penalty coefficient α.

Slow variable dimension D_s. The slow variable dimension is set from 1 to 5 respectively. The Koopman dimension is unified to 5. The MAPE of the predicted results is shown in Table 2.

Table 2: Performance of different D_s.

D_s	$\tau = 0.1$	$\tau = 1.0$	$\tau = 5.0$
1	0.59%	0.46%	0.16%
2	0.64%	0.48%	0.15%
3	0.64%	0.46%	0.14%
4	0.55%	0.45%	0.12%
5	0.57%	0.45%	0.13%
1*	2.68%	0.86%	0.18%
2*	1.36%	0.60%	0.15%
3*	1.36%	0.59%	0.12%
4*	1.32%	0.59%	0.11%
5*	1.33%	0.57%	0.09%

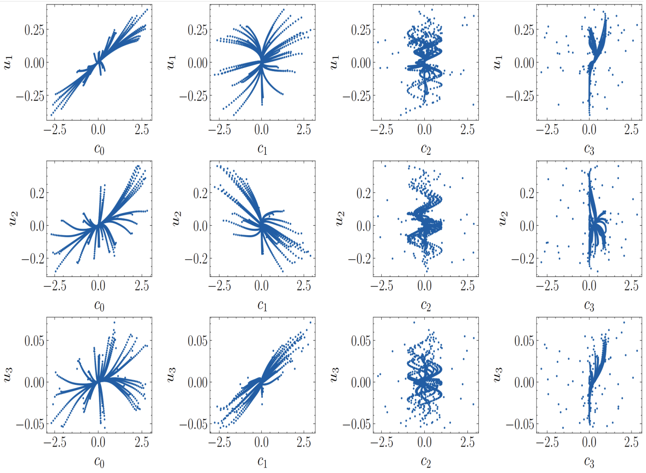


Figure 11: Scatter of representation and observed variables when $D_s = 3$.

When the D_s is set to 1, though the prediction performance is close to the other settings, the model can actually only identify one of the two slow variables (whose decay index is -1) at this point, and the other slow variable, i.e., c_2, is treated as a fast variable for prediction via autoregression, thus ensuring the robustness of the overall prediction performance. To verify this, we add a set of comparison experiments with the fast prediction module turned off, and the results show that there is a more significant drop in prediction performance when $D_s = 1$ than others. Therefore, when the number of slow variables set is less than the actual value, the slow variable cannot be extracted sufficiently, and the proposed framework fails to model the slow dynamics.

When the number of slow variables set is larger than the actual value, performance does not degrade, but it does not improve significantly either. In addition, the slow variables extracted by the model are mixed with redundant information and it is difficult to explain which one is the true slow variable representation at this point, as shown in Figure 11.

Koopman dimension K. The slow variable dimension is unified to 2, and Koopman dimensions are set from 2 to 6 in turn. The MAPE of the predicted results is shown in Table 3, where changes in the Koopman dimension do not significantly change the prediction performance.

Table 3: Performance of different K.

K	$\tau = 0.1$	$\tau = 1.0$	$\tau = 5.0$
2	0.61%	0.47%	0.16%
3	0.55%	0.46%	0.13%
4	0.60%	0.48%	0.11%
5	0.64%	0.48%	0.15%
6	0.57%	0.41%	0.10%

Penalty coefficient α. We test the impact of hyperparameter α in (6), which acts as a penalty coefficient constraining the encoder to extract only the slow variables, resulting in better long-time prediction results. Specifically, we evaluate the prediction performance of our proposed framework with α ranging from 0.1 to 0.5 in the 2S2F system, where $D_s = 2$ and $K = 2$. As shown in Table 4, the overall performance is best at $\alpha = 0.3$, but the performance does not change much when α takes other values. Therefore, the proposed model performance is robust to α.

Table 4: Performance of different α.

	$\tau = 0.1$		$\tau = 1.0$		$\tau = 5.0$	
α	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
0.1	0.0761	0.88%	0.0572	0.59%	0.0045	0.13%
0.2	0.0757	0.78%	0.0566	0.61%	0.0056	0.15%
0.3	0.0731	0.71%	0.0550	0.55%	0.0049	0.11%
0.4	0.0746	0.74%	0.0558	0.57%	0.0051	0.14%
0.5	0.0758	0.82%	0.0553	0.53%	0.0047	0.13%