Koopman Operator

Nazanin Bondarian

Machine Learning Course

Dr. Mehdi Aliyari

K. N. TOOSI University of Technology

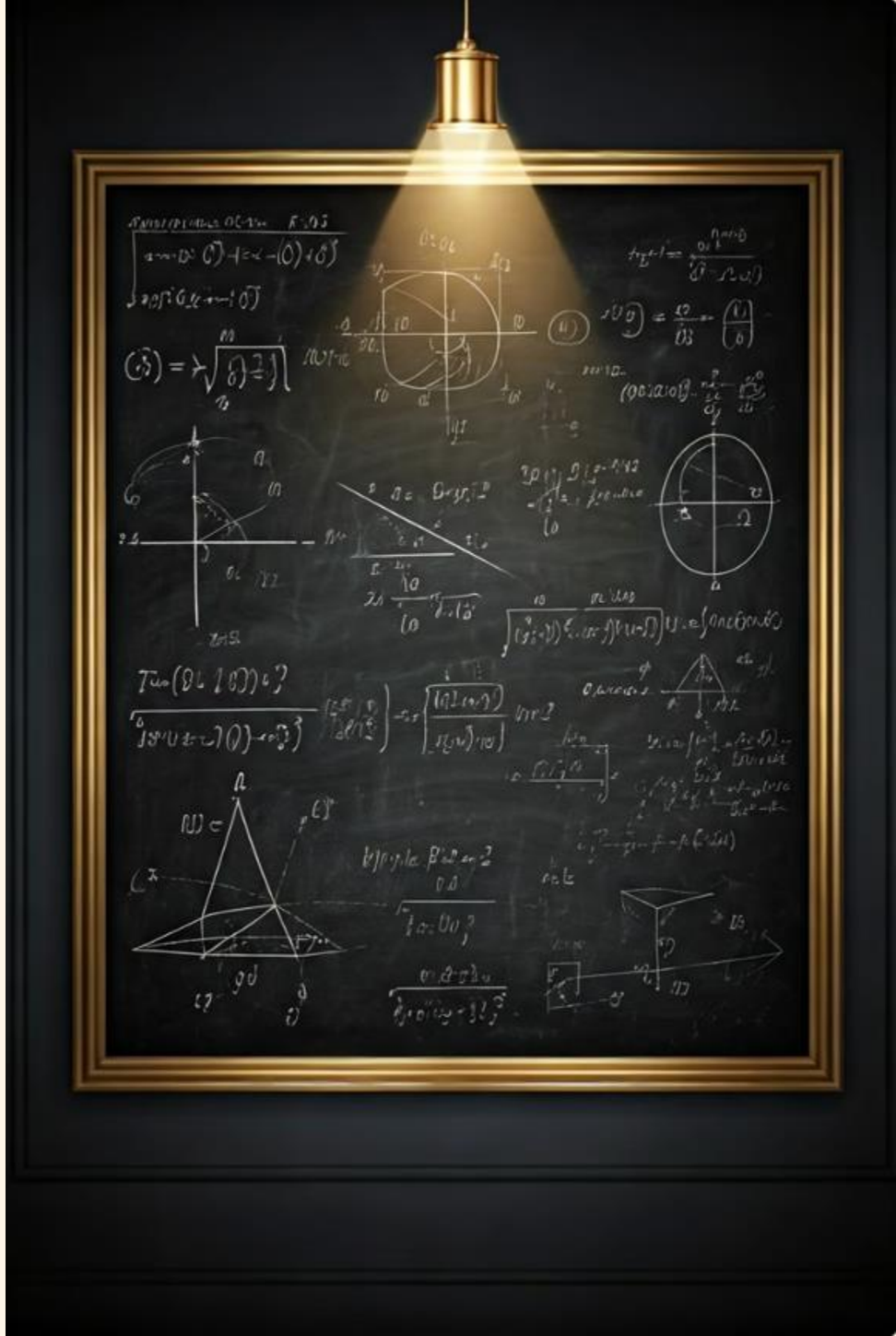
July 21, 2024

1928
K. N. Toosi University of Technology

Introduction to PyKoopman

PyKoopman is a powerful Python package that enables data-driven approximation of the Koopman operator, a crucial tool for understanding the dynamics of complex systems. This introductory section provides an overview of the package's capabilities and its potential applications in various scientific and engineering domains.

https://github.com/dynamicslab/pykoopman.



What is the Koopman Operator?

Linear Representation

The Koopman operator provides a linear representation of a nonlinear dynamical system, allowing for powerful analysis and prediction techniques.

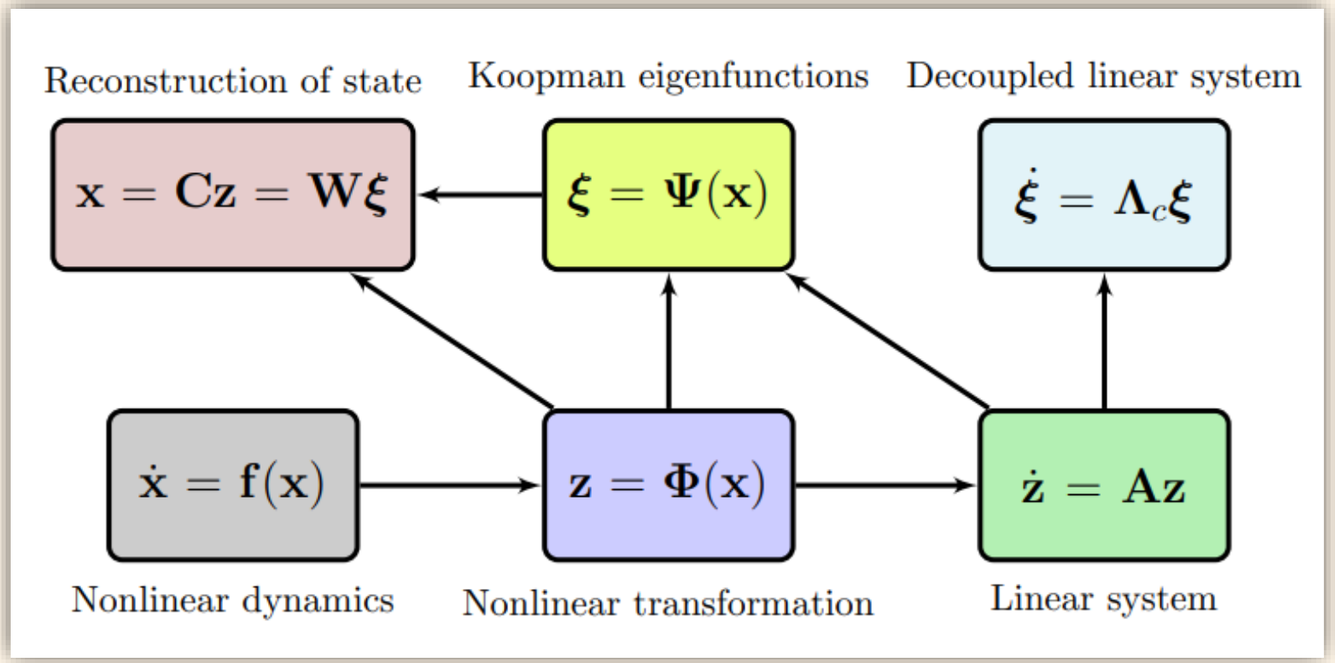
Infinite-Dimensional

The Koopman operator exists in an infinite-dimensional function space, capturing the evolution of all observables of the system.

Data-Driven Approximation

PyKoopman enables the data-driven approximation of the Koopman operator, making it accessible to a wide range of applications.

Koopman Operator



Lifting of the state x of the continuous autonomous dynamical system into a new coordinate system, in which the original nonlinear dynamics become linear and are easier to handle. One can also linearly reconstruct the state x from the new coordinate system. This is facilitated with PyKoopman in a data-driven manner.

Features of PyKoopman

1 Observables

The nonlinear observables used to lift x to z, and reconstruct x from z.

2 Regression:

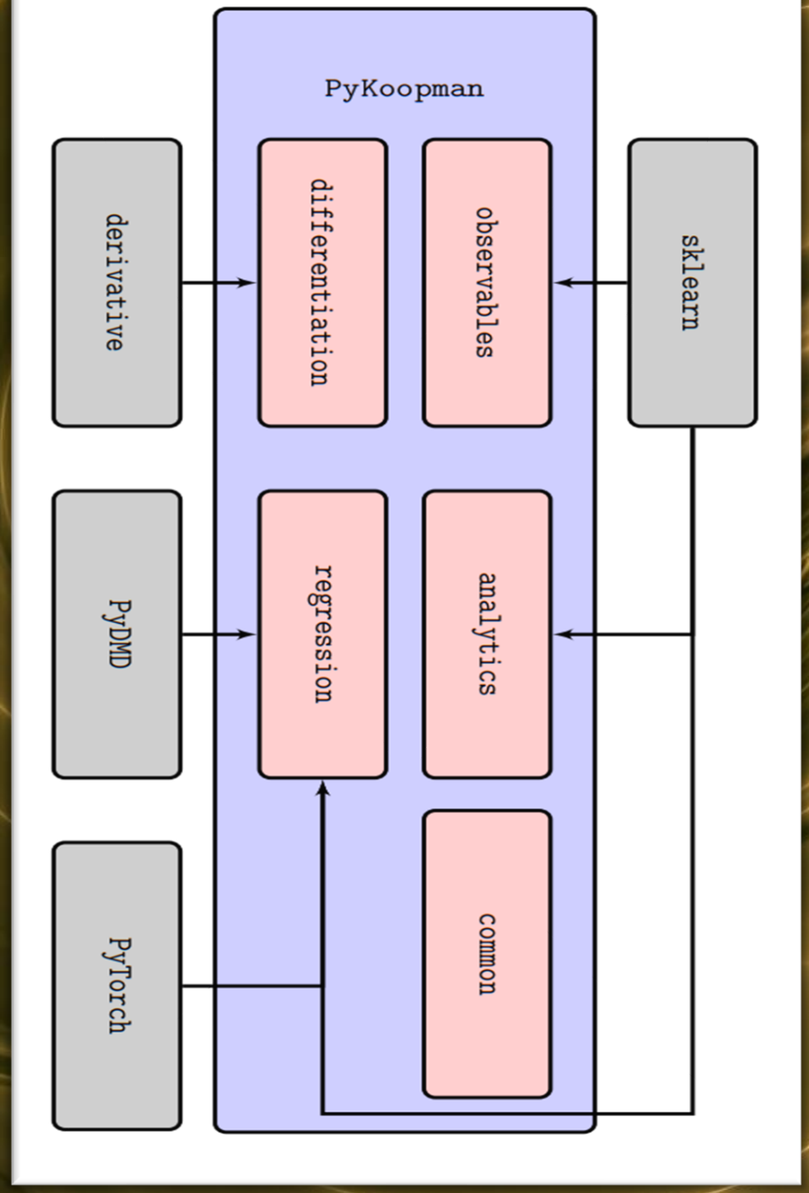
The regression used to find the best-fit dynamics operator A.

3 Differentiation

Evaluates the time derivative from a trajectory.

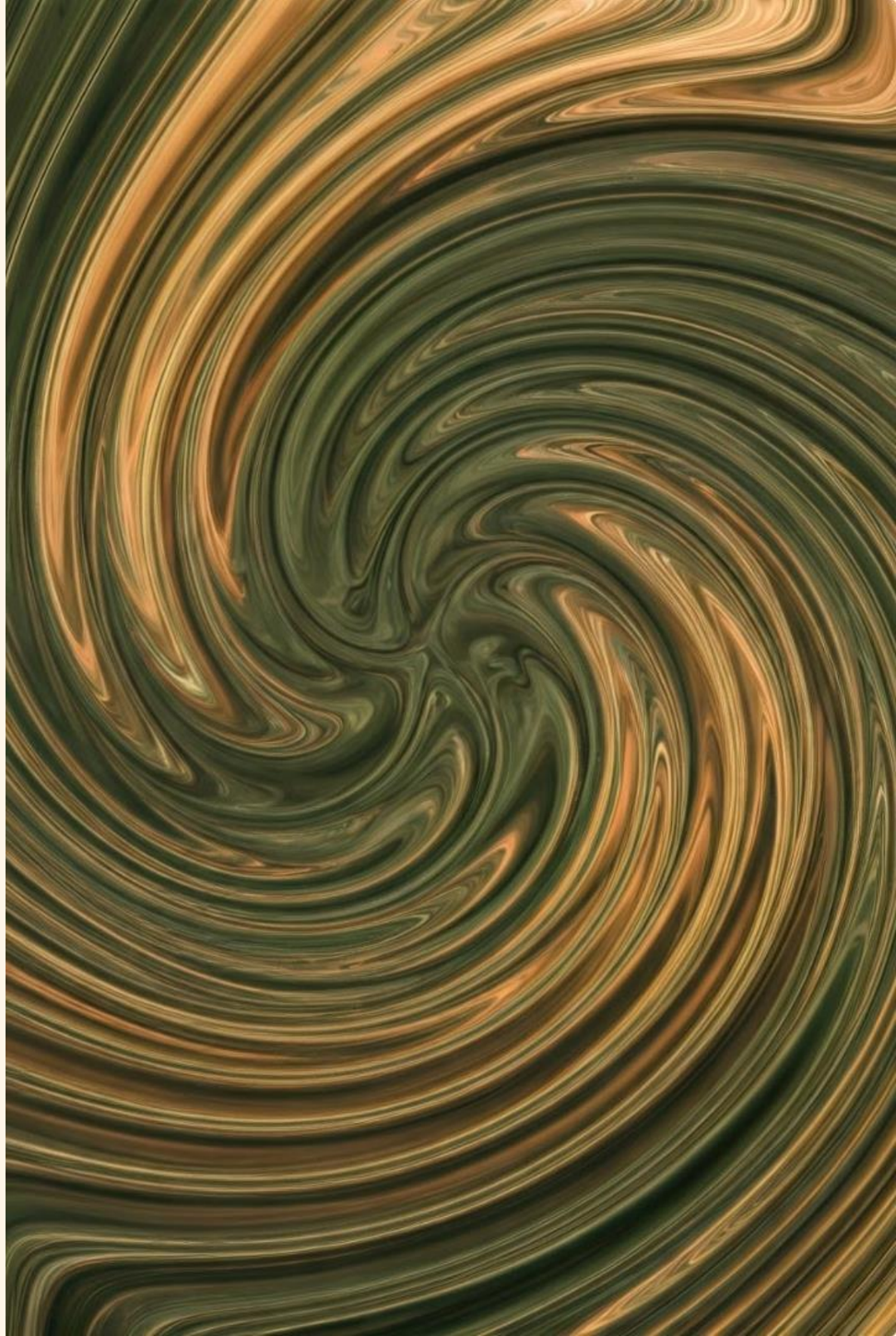
4 Analytics

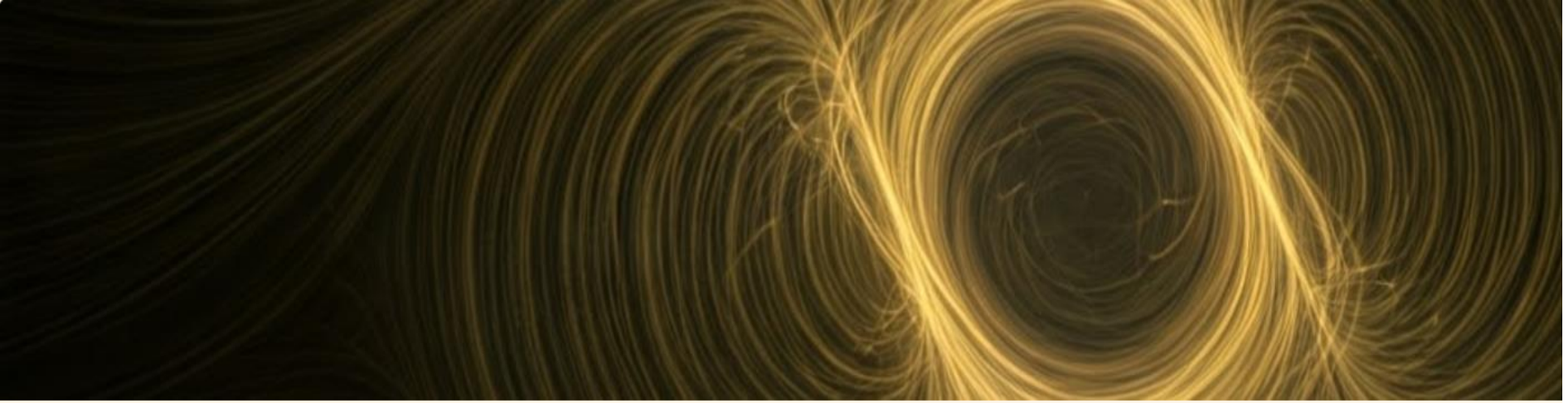
Sparsifying arbitrary approximations of the Koopman operator

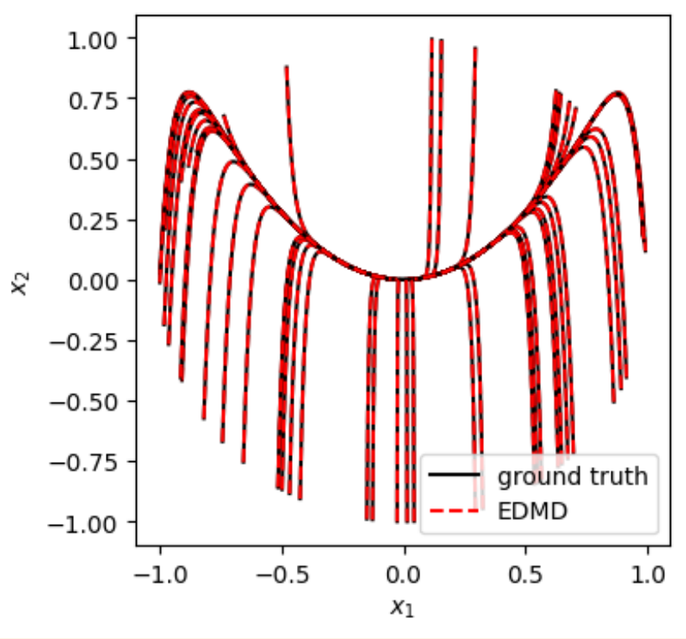


Features implemented

Observable library for lifting the state x into the observable space	Identity. Polynomial. TimeDelay. RadialBasisFunctions. RandomFourierFeatures. CustomObservables. ConcatObservables.
System identification method for performing regression	PyDMDRegressor. DMDc . EDMD. EDMDc. KDMD. HDMD. HDMDc. NNDMD







Example 1: Extended DMD for slow manifold

Data Generation

To prepare training data, we draw 100 random number within [−1,1] as initial conditions and then collect the corresponding trajectories forward in time:

1

Koopman Approximation

To begin, we create an observable function and an appropriate regressor. These two objects will then serve as input for the Koopman class. For instance, we employ EDMD to approximate the slow manifold dynamics.

2

Model Prediction

Use the model.simulate method to make predictions over an arbitrary time horizon..

3

Example 2: Different observables

Identity

The Identity observables simply leave the state variables unmodified.

Polynomial

Polynomial observables compute polynomial functions of the state variables.

Time-delays

It is often useful to use time-delayed versions of state variables.

Random Fourier Features

Custom observables

The CustomObservables class allows one to directly specify functions that should be applied to the state variables.

Concate observables

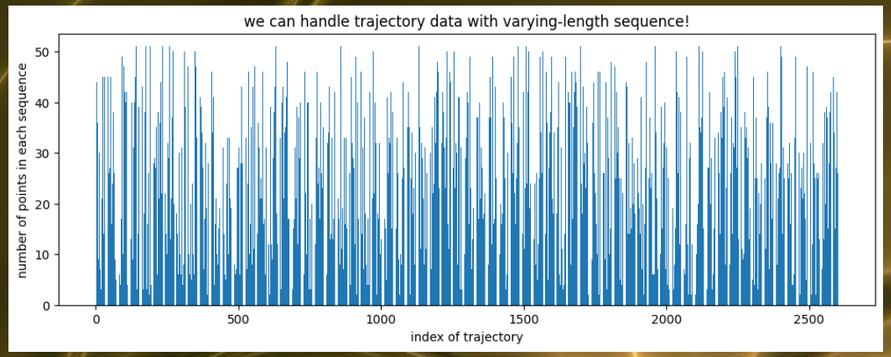
Example 3: Neural Network DMD on Slow Manifold

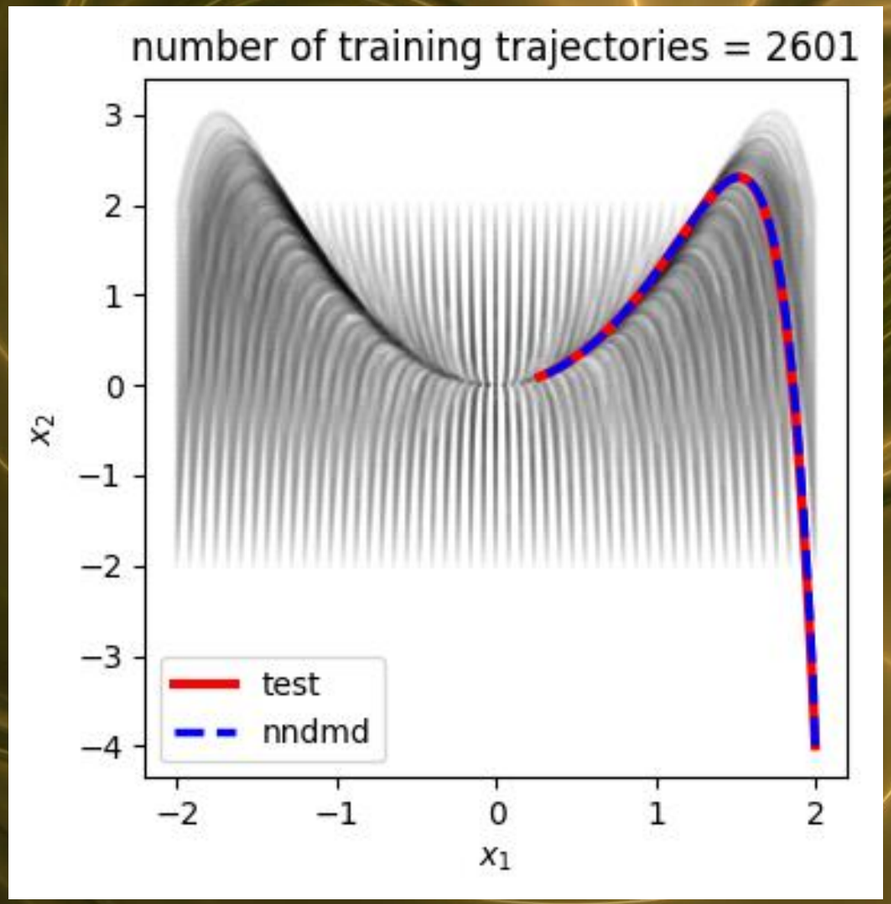
Collect training data

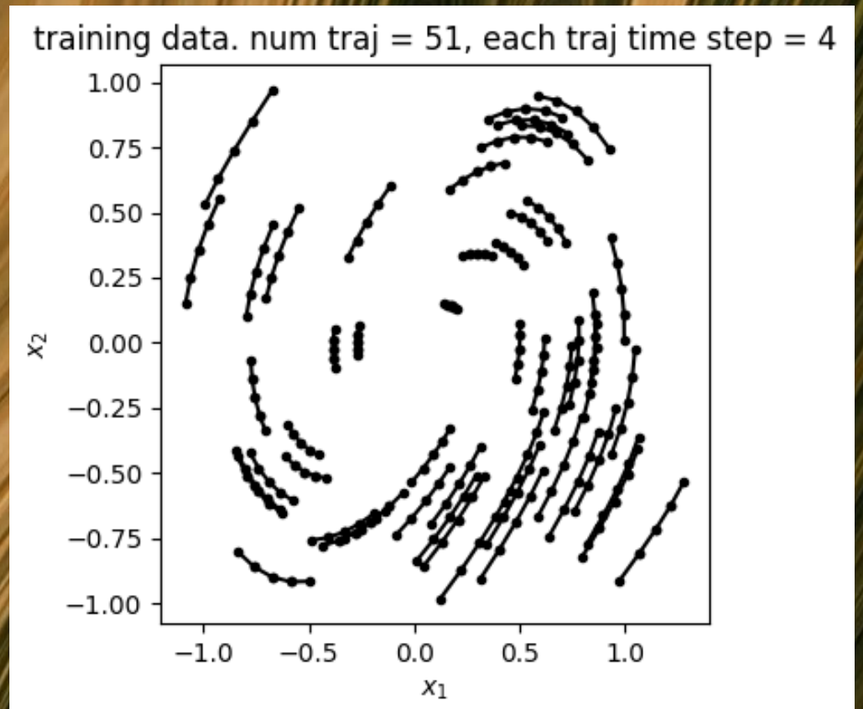
I will create a grid between -2 and 2 to sample initial conditions. I sample 51 points along each direction. Next, I will generate many sequence with different length.

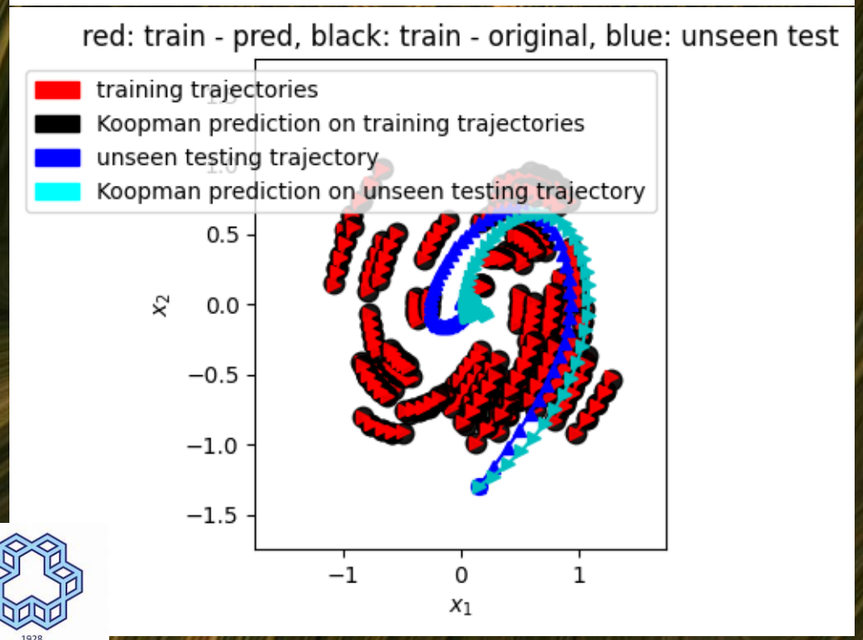
Call NNDMD as our model

Testing on unseen trajectory









Example 4: Extended DMD for Van Der Pol System

1

Collect training data

The training data consists of a snapshot pair taken from 51 uniformly distributed random initial conditions.

2

Koopman regression using EDMD

As basis functions we choose thinplate radial basis functions.

3

Prediction

Show the prediction performance for the training data.

Example 5: Extended DMD with control for Van der Pol oscillator

Training data

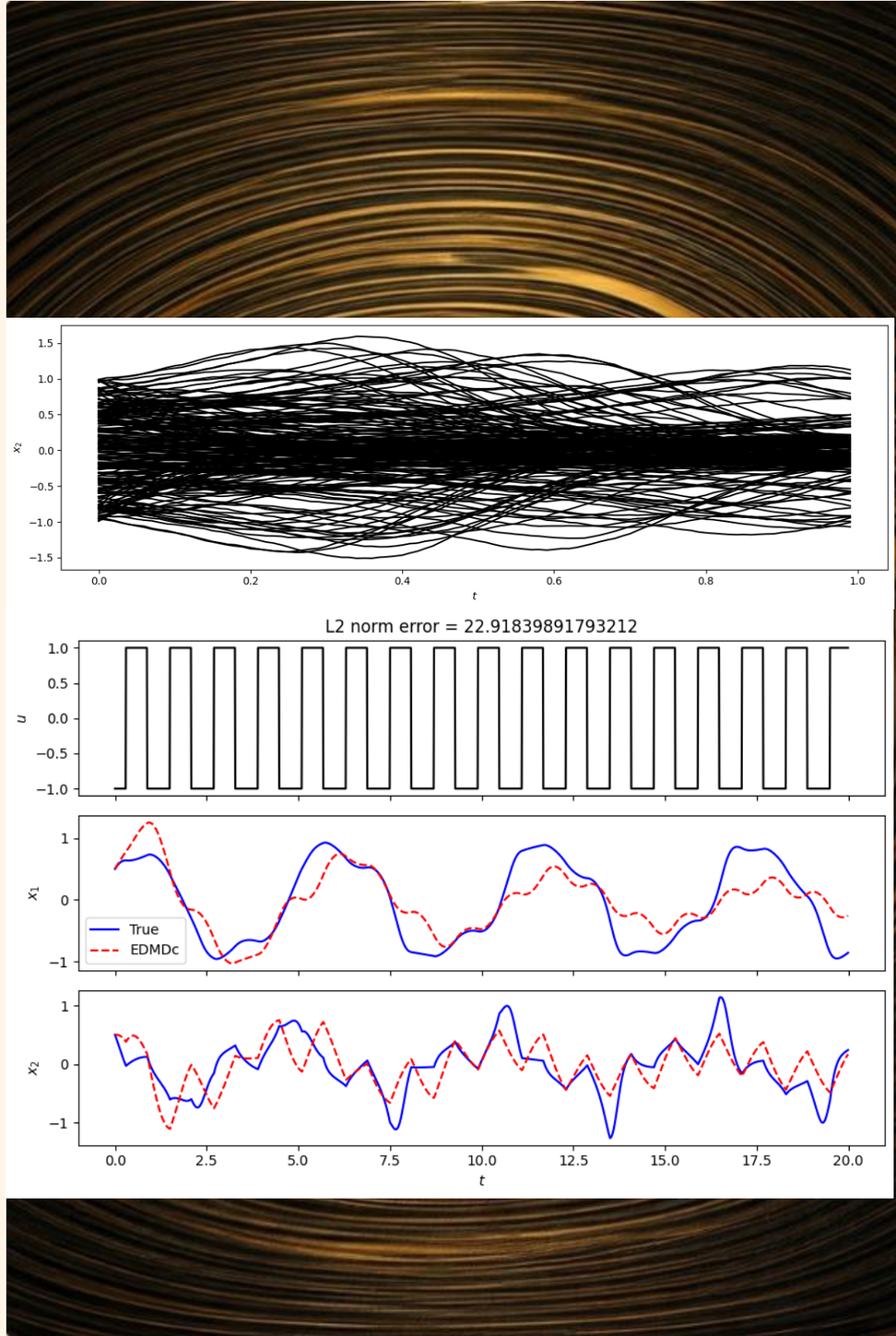
A training dataset is created consisting of 200 trajectories, each trajectory is integrated for 1000 timesteps.

EDMc Model

The observables for the Koopman model are chosen to be the state itself and thin plate radial basis functions with centers selected randomly.

Predict using Koopman model

The trained model is used to perform a multi-step prediction from a given initial condition.



Thanks for your attention!