

Empirical Dynamic Modelling

Automatic Causal Inference and Forecasting

Patrick Laub

UNSW Sydney, School of Risk and Actuarial Studies

February 7, 2023

Goal: automatic causal inference

```

df <- read.csv("chicago.csv")
head(df)
#>   Time Temperature Crime
#> 1    1        24.08  1605
#> 2    2        19.04  1119
#> 3    3        28.04  1127
#> 4    4        30.02  1154
#> 5    5        35.96  1251
#> 6    6        33.08  1276

library(fastEDM)

crimeCCMCausesTemp <- easy_edm("Crime", "Temperature", data=df)
#> ✗ No evidence of CCM causation from Crime to Temperature found.

tempCCMCausesCrime <- easy_edm("Temperature", "Crime", data=df)
#> ✓ Some evidence of CCM causation from Temperature to Crime found.

```

The Stata Journal (2021)
21, Number 1, pp. 220–258

DOI: 10.1177/1536867X211000030

Beyond linearity, stability, and equilibrium: The edm package for empirical dynamic modeling and convergent cross-mapping in Stata

Jinjing Li
University of Canberra

George Sugihara
University of California San Diego

Michael J. Zyphur
University of Queensland

Patrick J. Laub
UNSW



Acknowledgments

Discovery Project DP200100219 and Future Fellowship FT140100629.

Mirage correlation

$$X(t+1) = X(t) [3.8 - 3.8 X(t) - 0.02 Y(t)]$$

$$Y(t+1) = Y(t) [3.5 - 3.5 Y(t) - 0.1 X(t)]$$

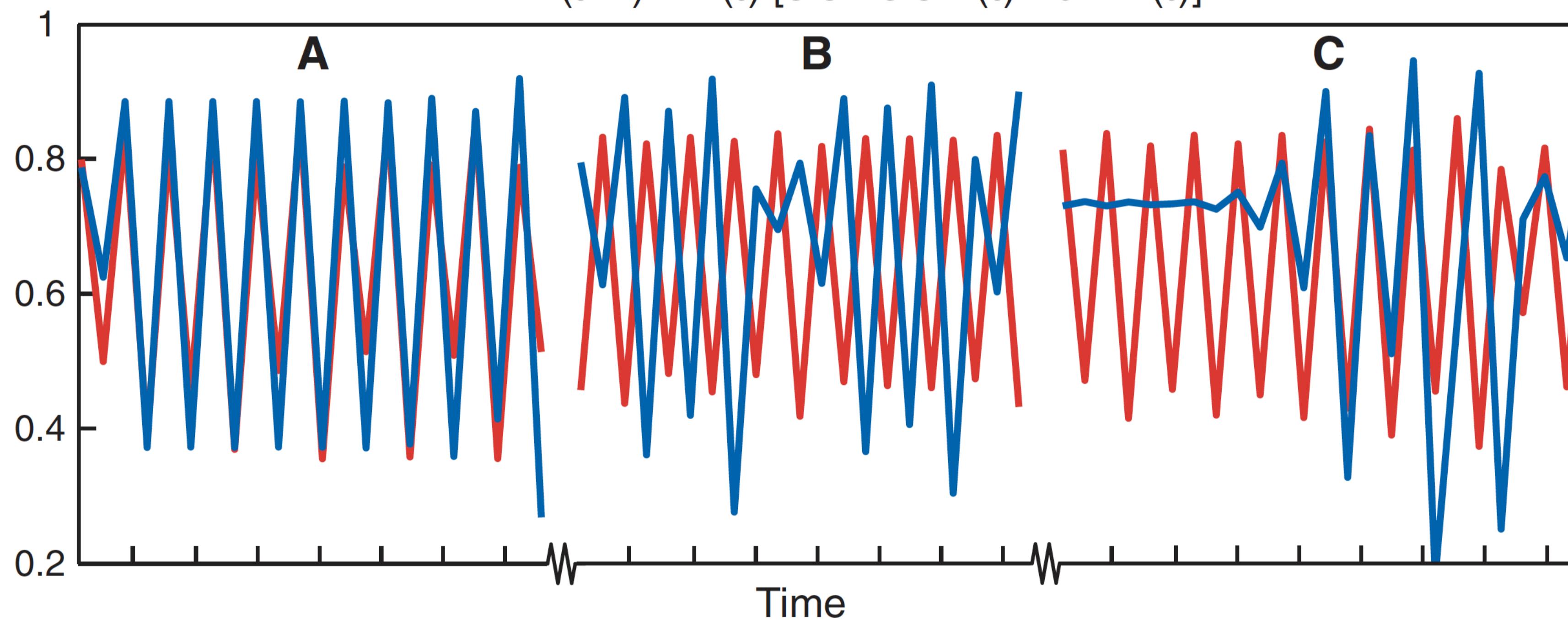


Fig. 1. Mirage correlations. (A to C) Three samples from a single run of a coupled two-species nonlinear logistic difference system with chaotic dynamics. Variables X (blue) and Y (red) appear correlated in the first time segment (A), anticorrelated in the second time segment (B), and lose all coherence in the third time segment (C) with alternating interspersed periods of positive, negative, and zero correlation. Although the system is deterministic and dynamically coupled, there is no long-term correlation ($n = 1000$, $\rho = 0.0054$, $P = 0.864$).

A different view of causality

Imagine x_t, y_t, z_t are interesting time series...

If the data is generated according to the nonlinear system:

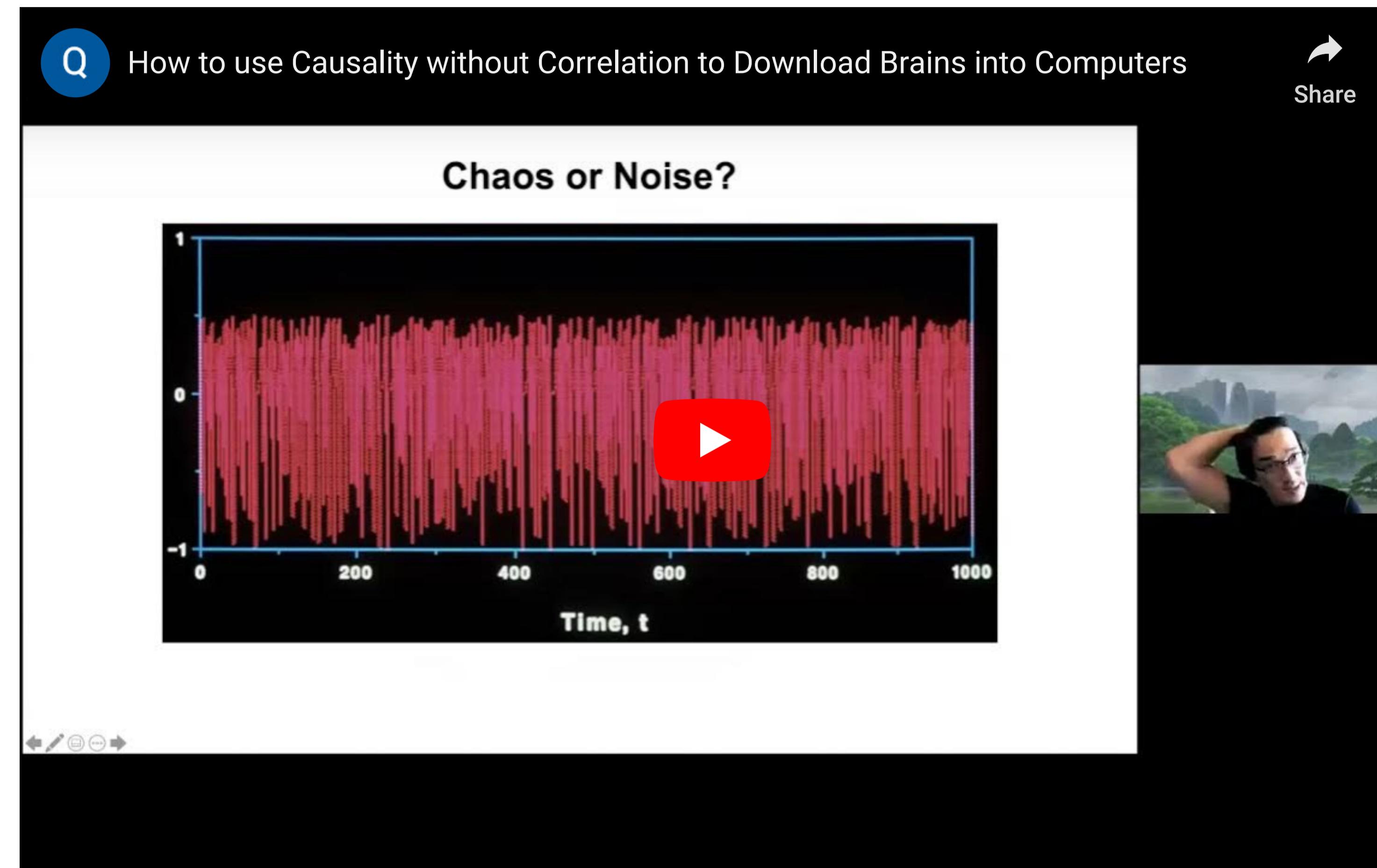
$$x_{t+1} = \sigma(y_t - x_t)$$

$$y_{t+1} = x_t(\rho - z_t) - y_t$$

$$z_{t+1} = x_t y_t - \beta z_t$$

then $y \Rightarrow x$, both $x, z \Rightarrow y$, and both $x, y \Rightarrow z$.

Fish brains



Gerald Pao (2021), “How to use Causality without Correlation to Download Brains into Computers”, QMNET Seminar.

Linear/nonlinear dynamical systems

Say $\mathbf{x}_t = (x_t, y_t, z_t)$, then if:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t$$

we have a linear system.

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t)$$

we have a nonlinear system.

Using a term like nonlinear science is like referring to the bulk of zoology as the study of non-elephant animals. (Stanisław Ulam)

Unobserved variables

Takens' theorem to the rescue, though...

Takens' theorem is a deep mathematical result with far-reaching implications. Unfortunately, to really understand it, it requires a background in topology.
 (Munch et al. 2020)

Takens's theorem

From Wikipedia, the free encyclopedia

In the study of [dynamical systems](#), a **delay embedding theorem** gives the conditions under which a [chaotic dynamical system](#) can be reconstructed from a sequence of observations of the state of a dynamical system. The reconstruction preserves the properties of the dynamical system that do not change under smooth coordinate changes (i.e., [diffeomorphisms](#)), but it does not preserve the geometric shape of structures in phase space.

Simplified, slightly inaccurate version [\[edit\]](#)

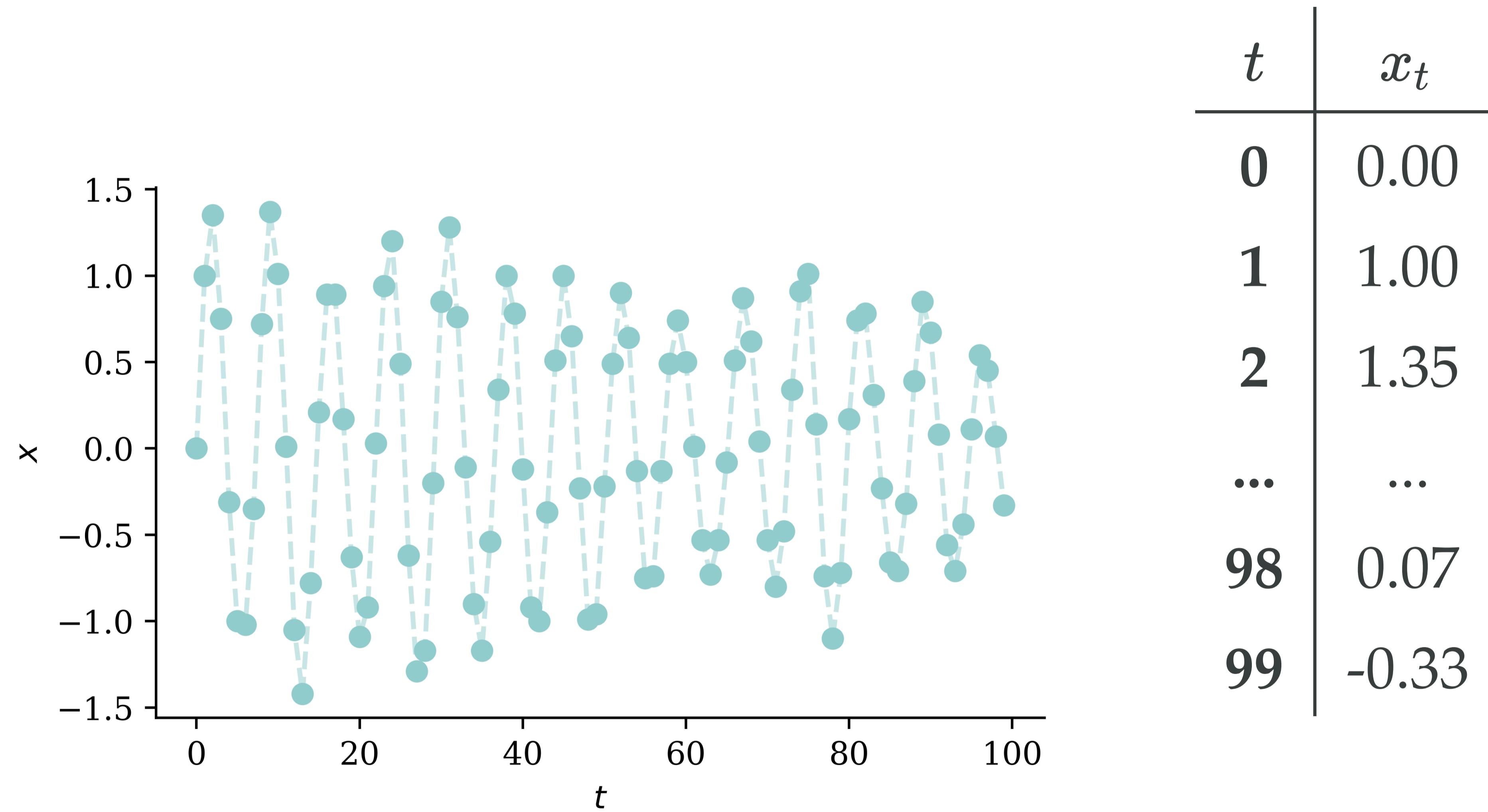
Suppose the d -dimensional state vector x_t evolves according to an unknown but continuous and (crucially) deterministic dynamic. Suppose, too, that the one-dimensional observable y is a smooth function of x , and "coupled" to all the components of x . Now at any time we can look not just at the present measurement $y(t)$, but also at observations made at times removed from us by multiples of some lag $\tau : y_{t+\tau}, y_{t+2\tau}$, etc. If we use k lags, we have a k -dimensional vector. One might expect that, as the number of lags is increased, the motion in the lagged space will become more and more predictable, and perhaps in the limit $k \rightarrow \infty$ would become deterministic. In fact, the dynamics of the lagged vectors become deterministic at a finite dimension; not only that, but the deterministic dynamics are completely equivalent to those of the original state space (More exactly, they are related by a smooth, invertible change of coordinates, or diffeomorphism.) The magic embedding dimension k is at most $2d + 1$, and often less.^[1]

Contents [\[hide\]](#)

- [1 Simplified, slightly inaccurate version](#)
- [2 See also](#)
- [3 References](#)
- [4 Further reading](#)
- [5 External links](#)

Simplex Algorithm

Toy problem

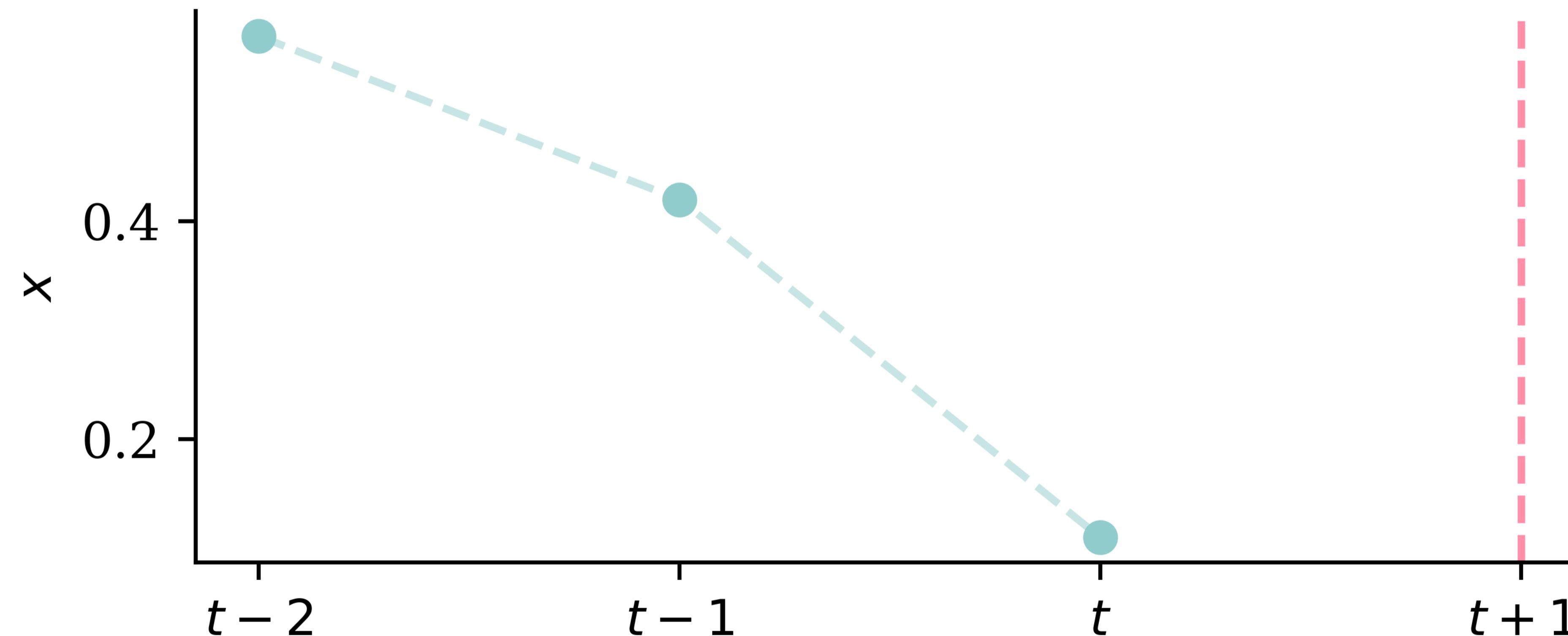


Create the embeddings

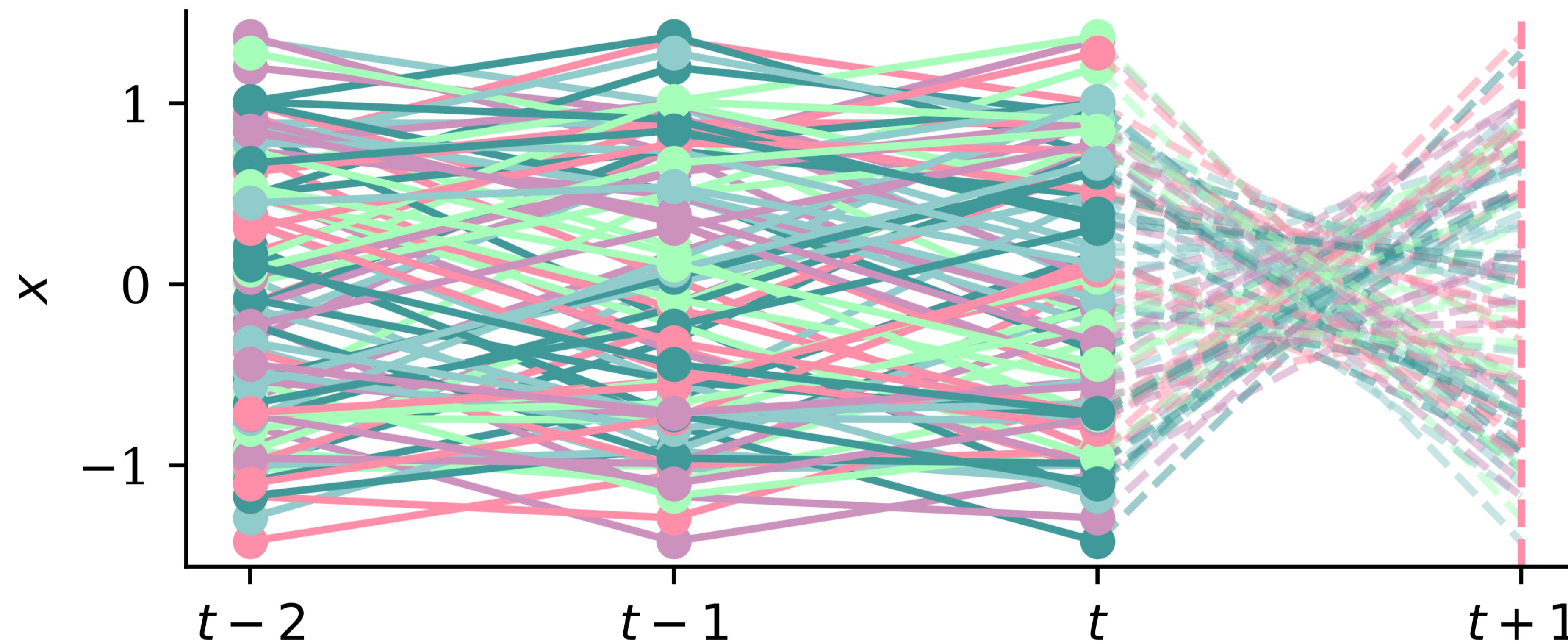
t	x_t	i	x_t	x_{t-1}	x_{t-2}	i	x_{t+1}
0	0.00	0	1.35	1.00	0.00	0	0.75
1	1.00	1	0.75	1.35	1.00	1	-0.31
2	1.35	2	-0.31	0.75	1.35	2	-1.00
...
98	0.07	95	0.45	0.54	0.11	95	0.07
99	-0.33	96	0.07	0.45	0.54	96	-0.33

Making a prediction

$$\boldsymbol{x}^* = (0.11, 0.42, 0.57)$$



Look at the manifold



Calculate the distances to the point

i	x_t	x_{t-1}	x_{t-2}	x_{t+1}	$d(\mathbf{x}_i, \mathbf{x}^*)$
0	1.35	1.00	0.00	0.75	0.98
1	0.75	1.35	1.00	-0.31	1.30
2	-0.31	0.75	1.35	-1.00	1.56
...
95	0.45	0.54	0.11	0.07	0.17
96	0.07	0.45	0.54	-0.33	0.66

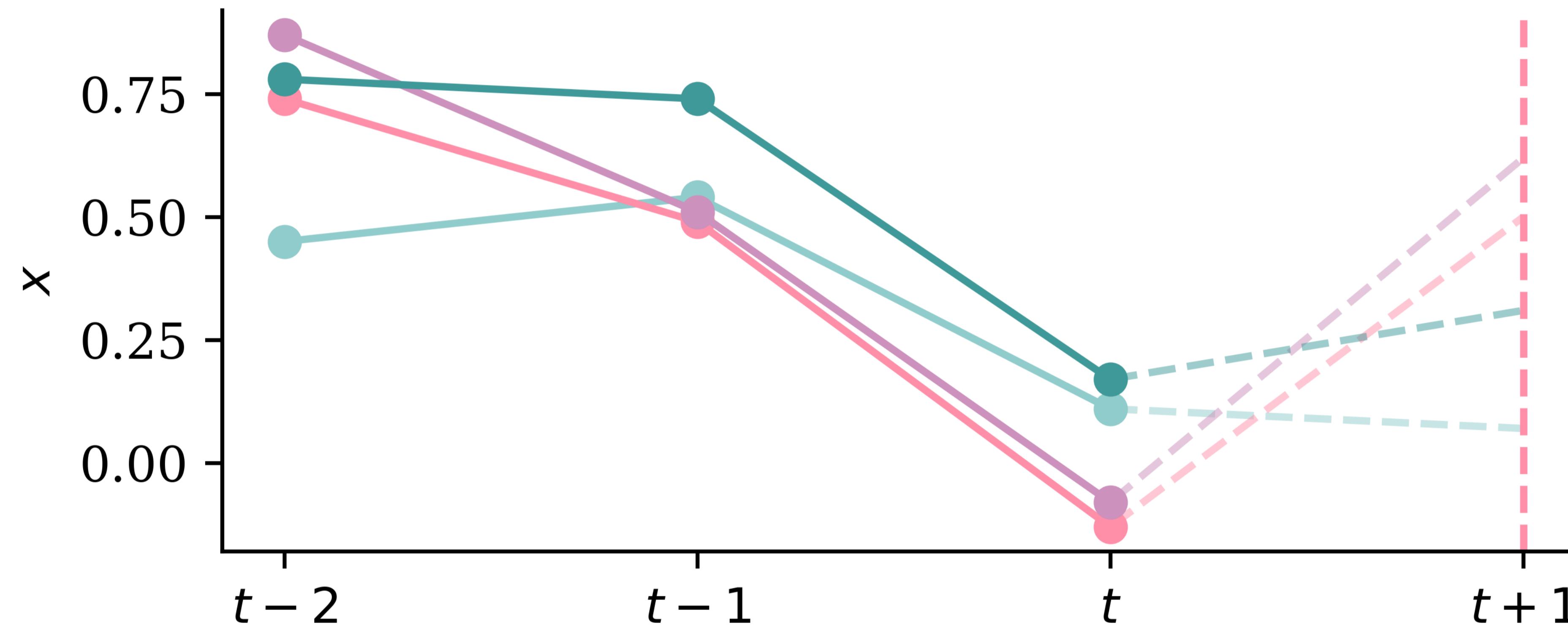
Find the nearest neighbours

i	x_t	x_{t-1}	x_{t-2}	x_{t+1}	$d(\mathbf{x}_i, \mathbf{x}^*)$
95	0.45	0.54	0.11	0.07	0.17
57	0.74	0.49	-0.13	0.50	0.30
65	0.87	0.51	-0.08	0.62	0.37
...
26	-1.17	-1.29	-0.62	-0.20	2.55
12	-0.78	-1.42	-1.05	0.21	2.56

Find the nearest neighbours

i	x_t	x_{t-1}	x_{t-2}	x_{t+1}	$d(\mathbf{x}_i, \mathbf{x}^*)$
95	0.45	0.54	0.11	0.07	0.17
57	0.74	0.49	-0.13	0.50	0.30
65	0.87	0.51	-0.08	0.62	0.37
80	0.78	0.74	0.17	0.31	0.39

Plot those trajectories

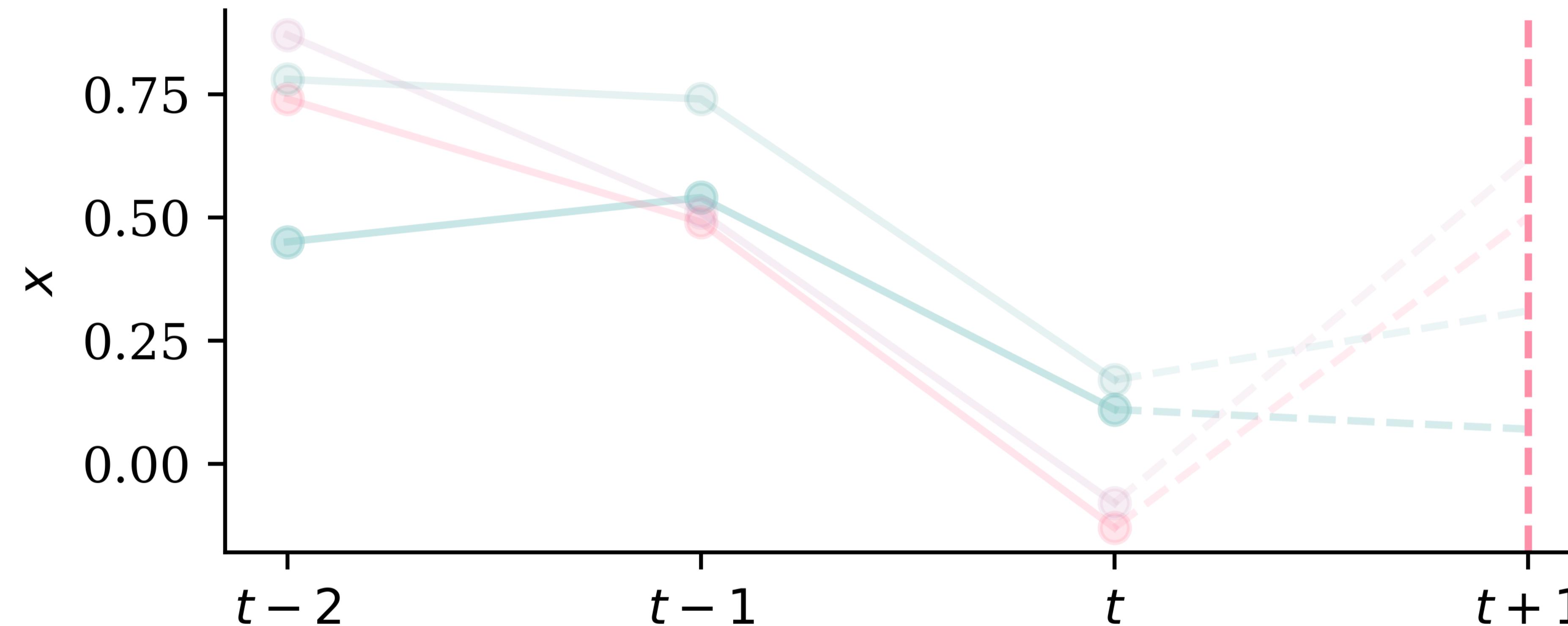


Make a prediction

i	x_{t+1}	$d(\mathbf{x}_i, \mathbf{x}^*)$	$\tilde{d}(\mathbf{x}_i, \mathbf{x}^*)$	w_i
95	0.07	0.17	1.00	0.49
57	0.50	0.30	1.76	0.23
65	0.62	0.37	2.18	0.15
80	0.31	0.39	2.29	0.13

$$w_i = \frac{\exp\{-\theta \tilde{d}(\mathbf{x}_i, \mathbf{x}^*)\}}{\sum_{j=1}^k \exp\{-\theta \tilde{d}(\mathbf{x}_j, \mathbf{x}^*)\}}$$

Plot those trajectories (weighted)

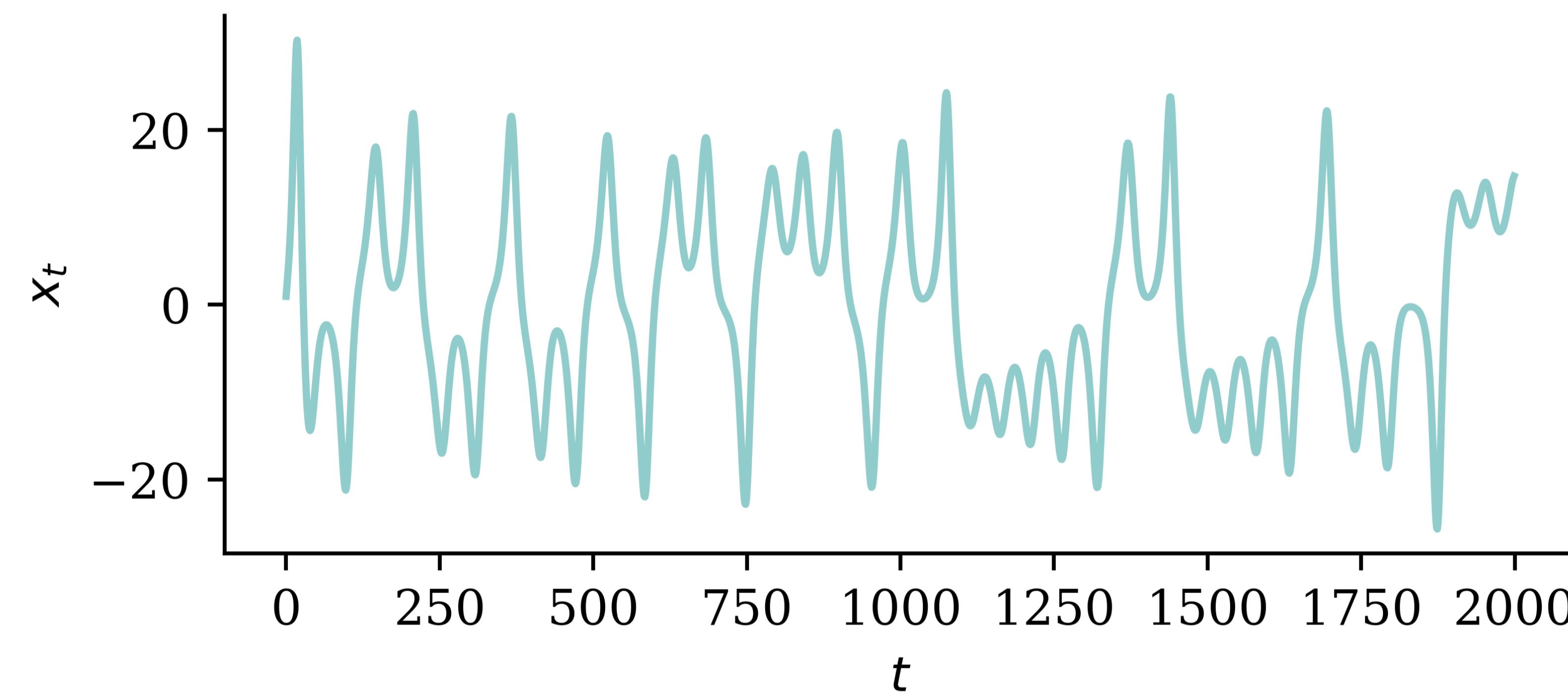


S-map Algorithm

Sequential Locally Weighted Global Linear Maps

Lorenz system

Given x_t from the Lorenz system:



predict x_{t+10} given (x_t, x_{t-20}) .

Time series to embedding

t	x_t	i	x_t	x_{t-20}	i	x_{t+10}
0	0.95	0	27.53	0.95	0	-4.75
1	1.85	1	24.75	1.85	1	-6.82
2	2.75	2	21.34	2.75	2	-8.63
...
1998	14.52	1768	-16.59	-4.76	1768	-15.64
1999	14.69	1769	-17.32	-4.90	1769	-14.56

Calculate normalised distances

i	x_t	x_{t-20}	x_{t+10}	$d(\mathbf{x}_i, \mathbf{x}^*)$	$\hat{d}(\mathbf{x}_i, \mathbf{x}^*)$
0	27.53	0.95	-4.75	18.32	0.91
1	24.75	1.85	-6.82	15.45	0.77
2	21.34	2.75	-8.63	12.06	0.60
...
1768	-16.59	-4.76	-15.64	31.18	1.55
1769	-17.32	-4.90	-14.56	31.90	1.58

The average of the distances is:

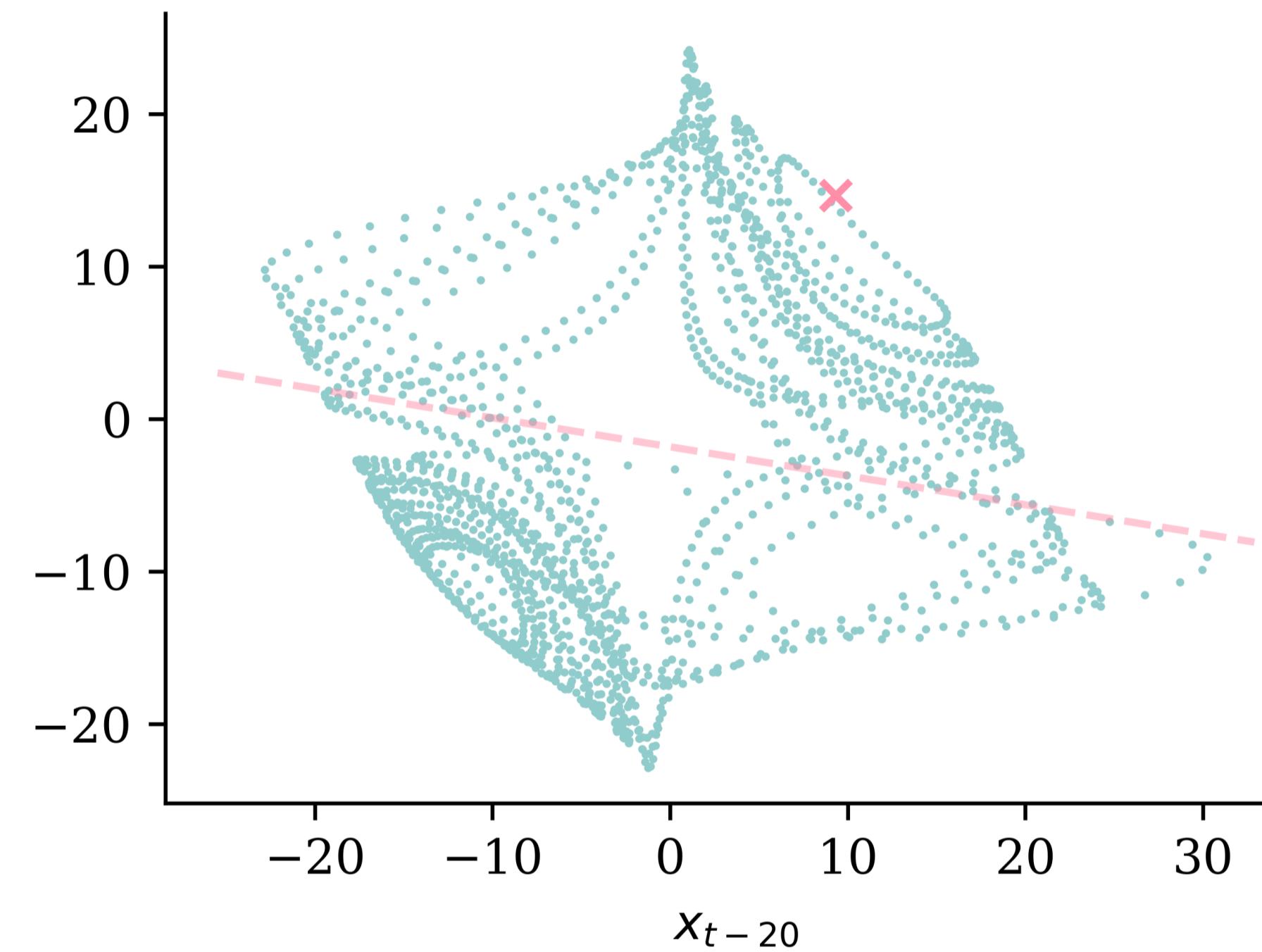
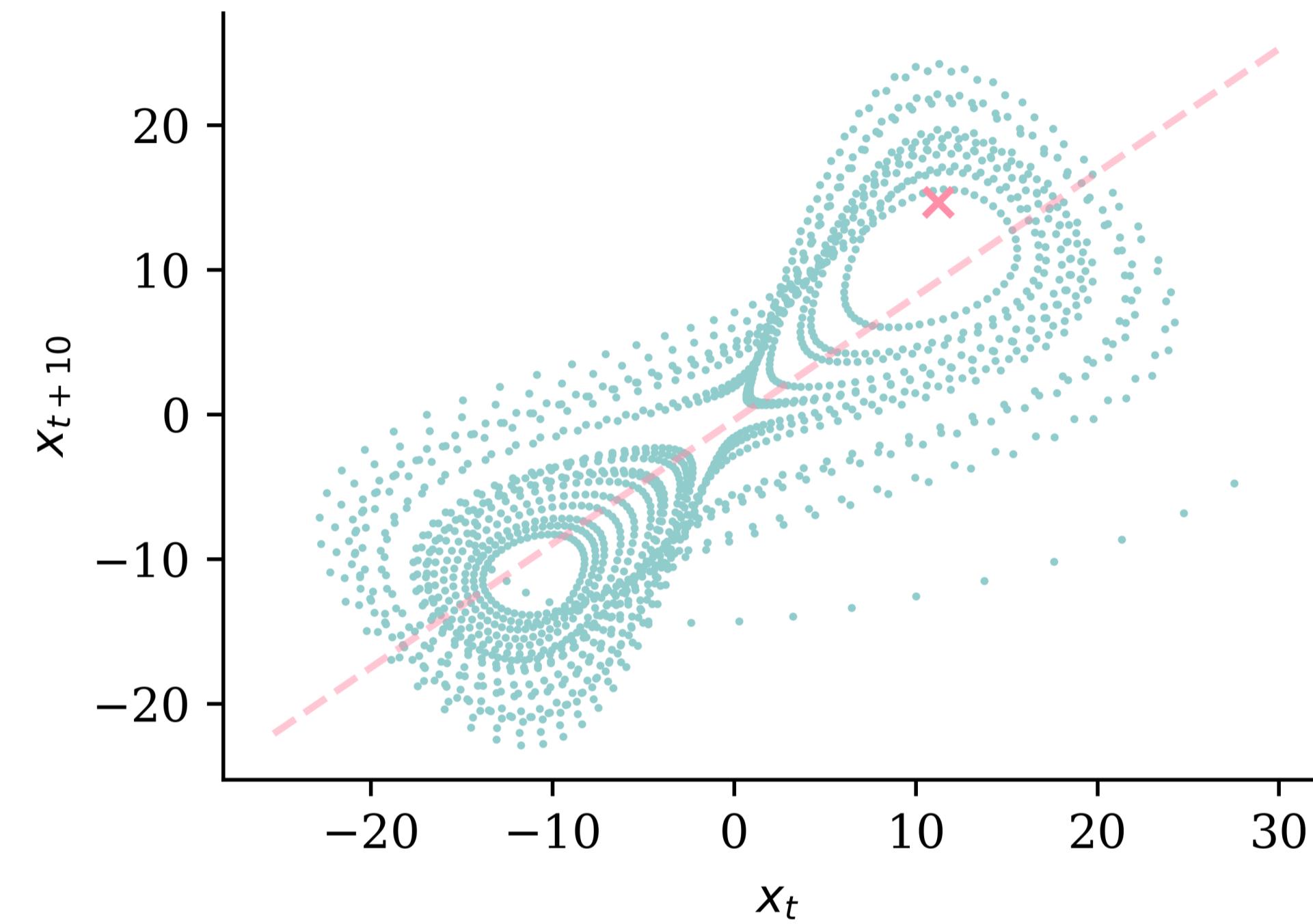
20.13

Weights

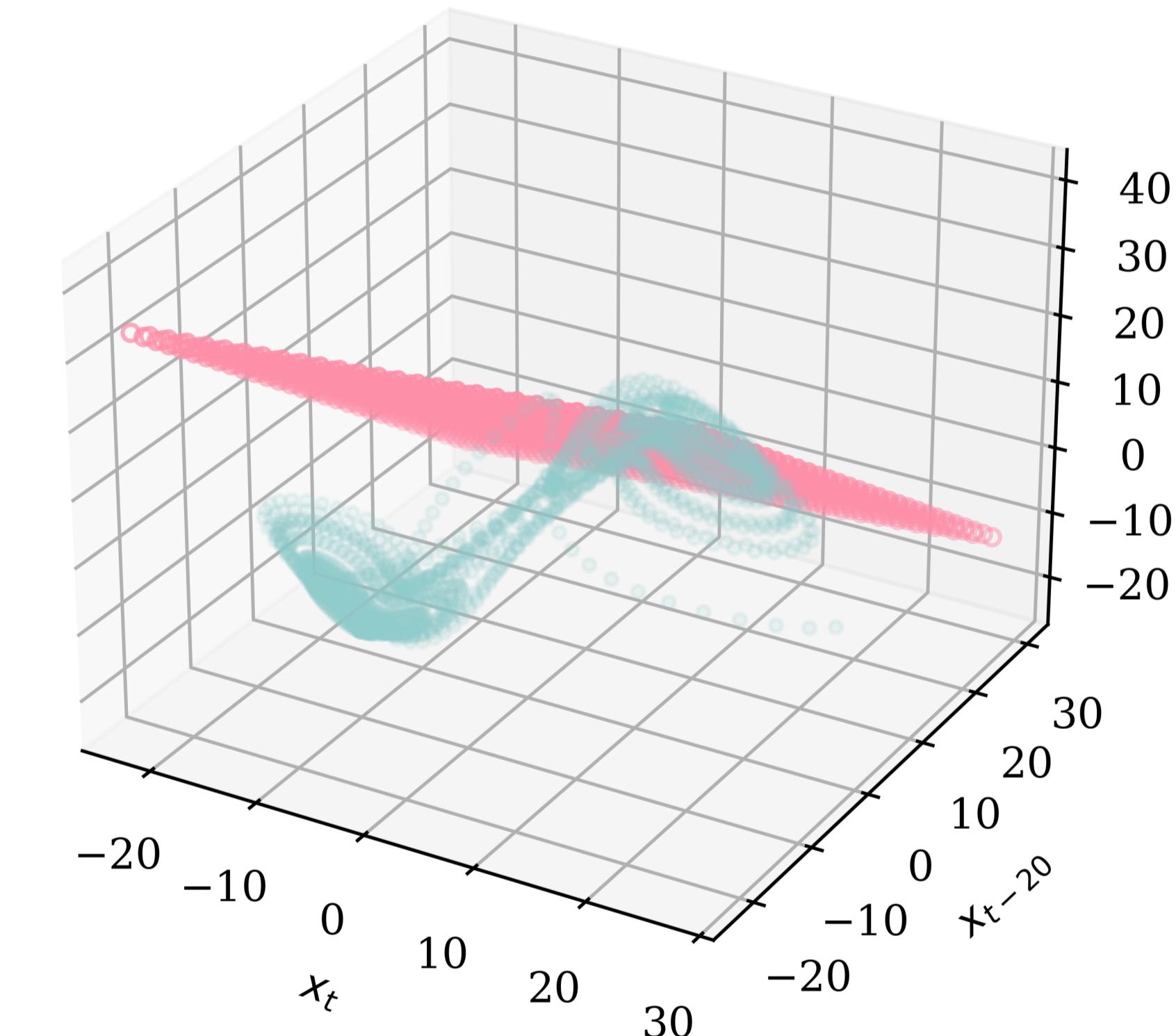
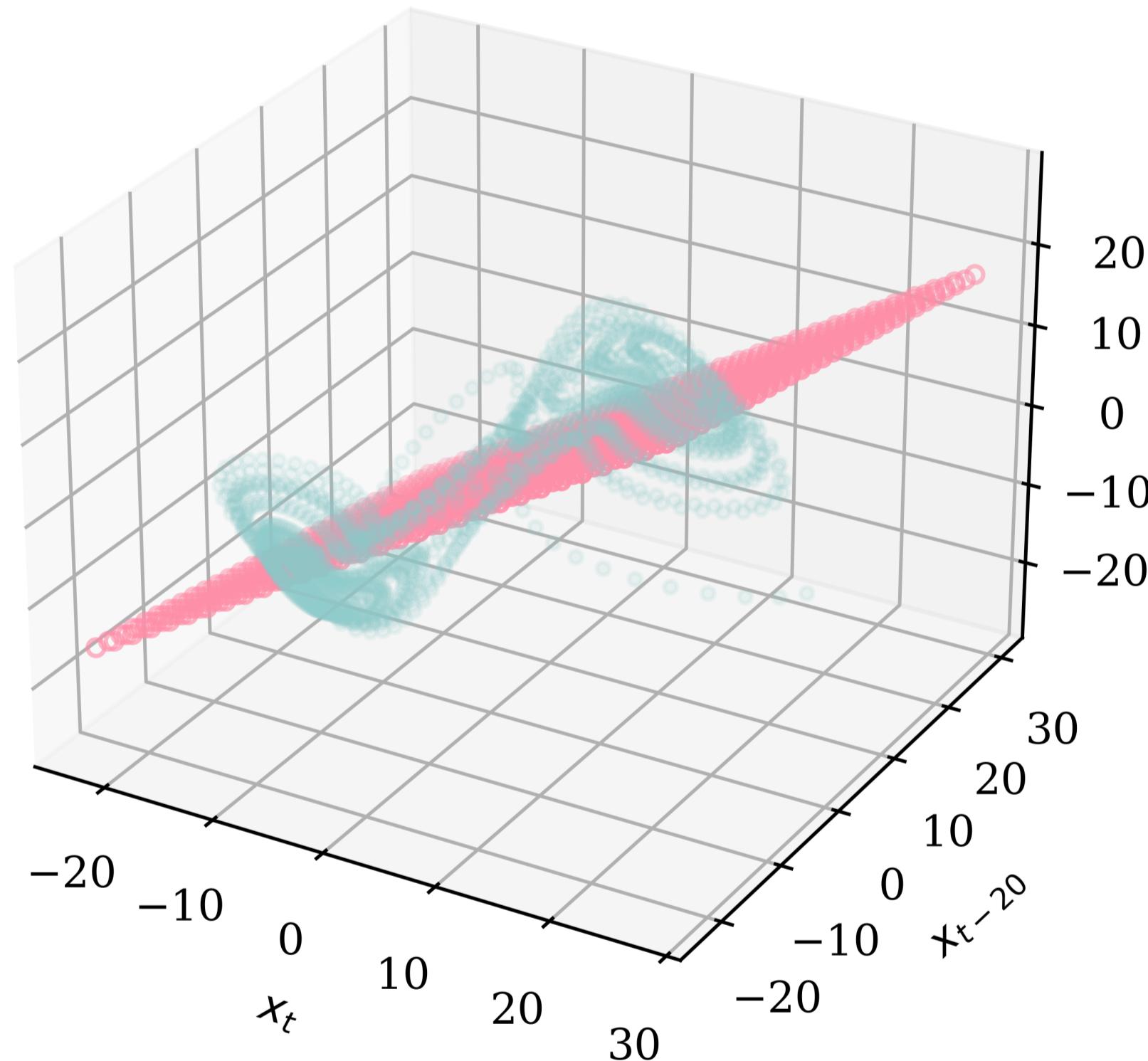
i	x_t	x_{t-20}	$d(\mathbf{x}_i, \mathbf{x}^*)$	w_i
0	27.53	0.95	18.32	0.01
1	24.75	1.85	15.45	0.02
2	21.34	2.75	12.06	0.05
...
1768	-16.59	-4.76	31.18	0.00
1769	-17.32	-4.90	31.90	0.00

$$w_i = \exp\{-\theta \hat{d}(\mathbf{x}_i, \mathbf{x}^*)\}$$

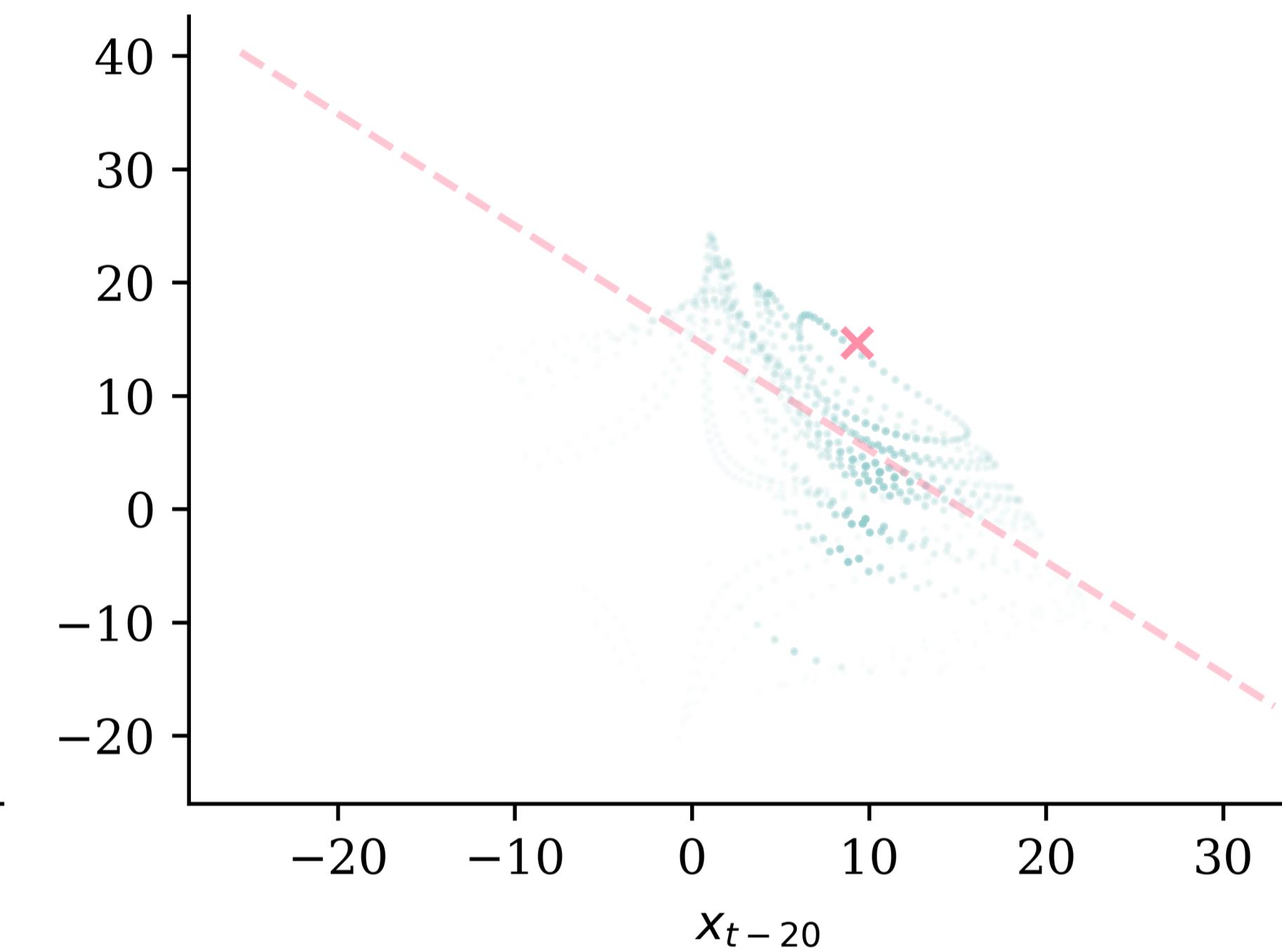
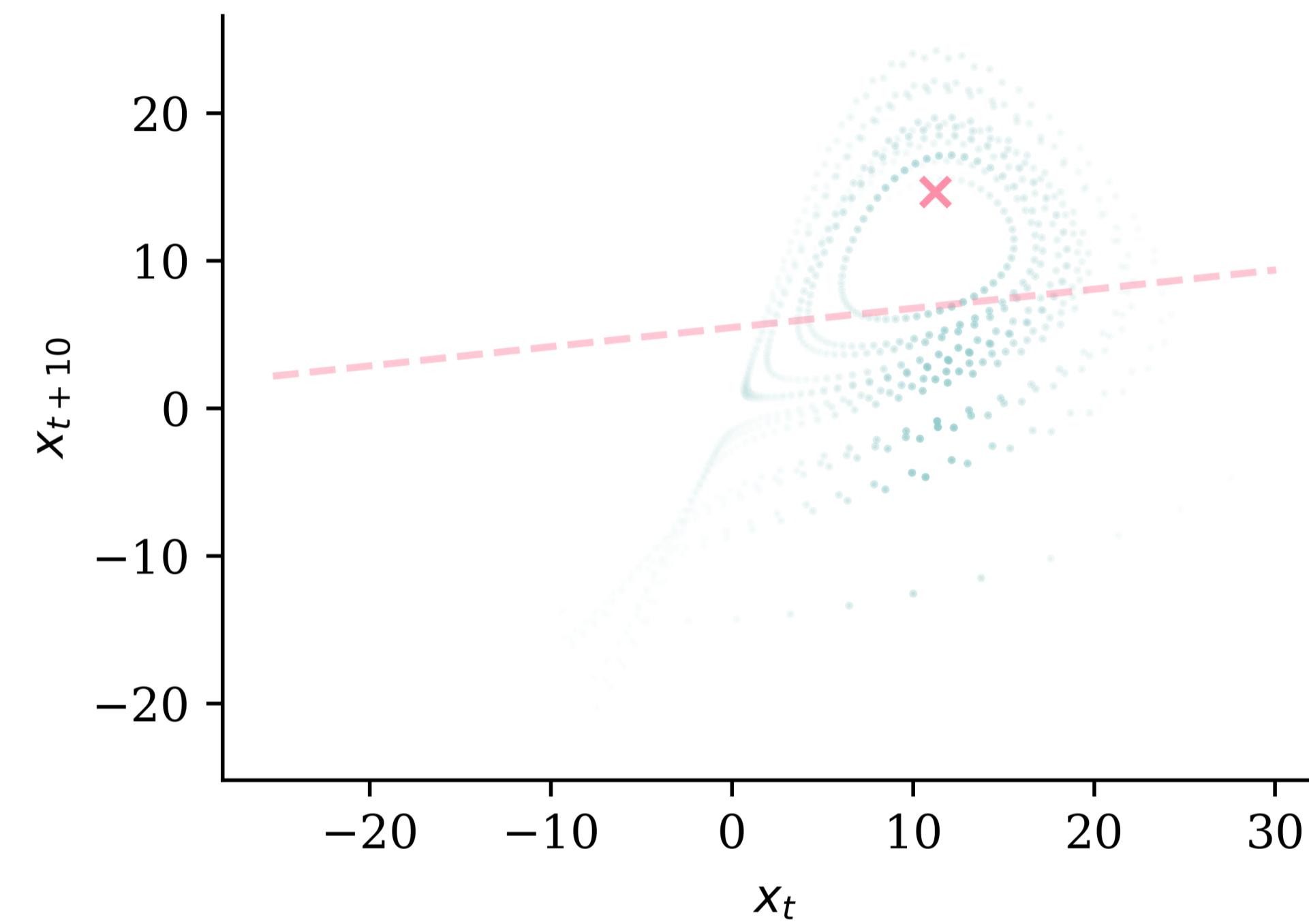
Linear auto-regression



Linear regression plane



Weighted linear regression



Empirical Dynamic Modelling (EDM)

Create lagged embeddings

Given two time series, create E -length trajectories

$$\mathbf{x}_t = (\text{Temp}_t, \text{Temp}_{t-1}, \dots, \text{Temp}_{t-(E-1)}) \in \mathbb{R}^E$$

and targets

$$y_t = \text{Crime}_t.$$

 **Note**

The \mathbf{x}_t 's are called *points* (on the shadow manifold).

Split the data

- $\mathcal{L} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ is *library set*,
- $\mathcal{P} = \{(\mathbf{x}_{n+1}, y_{n+1}), \dots, (\mathbf{x}_T, y_T)\}$ is *prediction set*.

For point $\mathbf{x}_s \in \mathcal{P}$, pretend we don't know y_s and try to predict it.

$$\forall \mathbf{x} \in \mathcal{L} \quad \text{find} \quad d(\mathbf{x}_s, \mathbf{x})$$

This is computationally demanding.

Convergent cross mapping

- If Temp_t causes Crime_t , then information about Temp_t is somehow embedded in Crime_t .
- By observing Crime_t , we should be able to forecast Temp_t .
- By observing more of Crime_t (more “training data”), our forecasts of Temp_t should be more accurate.

Example: Chicago crime and temperature.

Software

Stata package

≡ EDM Stata Package

 Search

Empirical Dynamic Modeling Stata Package

Package Description

Empirical Dynamic Modeling (EDM) is a way to perform *causal analysis on time series data*. The `edm` Stata package implements a series of EDM tools, including the convergent cross-mapping algorithm.

Key features of the package:

- powered by a fast multi-threaded *C++ backend*,
- able to process panel data, a.k.a. *multipatial EDM*,
- able to handle *missing data* using new `dt` algorithms or by dropping points,
- *factor variables* can be added to the analysis,
- *multiple distance functions* available (Euclidean, Mean Absolute Error, Wasserstein),
- *GPU acceleration* available.

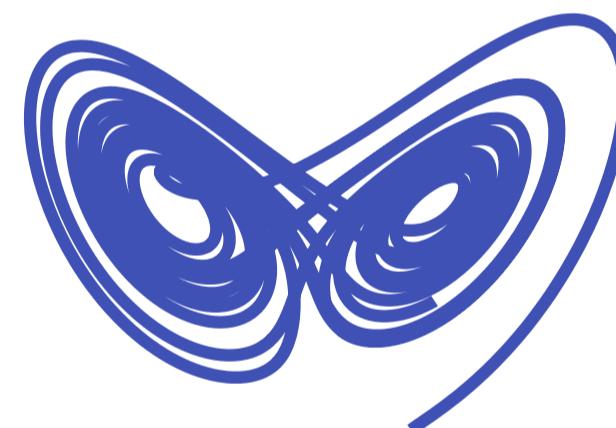


Table of contents

[Package Description](#)

[Installation](#)

[R & Python packages](#)

[Other Resources](#)

[Authors](#)

[Citation](#)

R package

Thanks to Rishi Dhushiyandan for his hard work on [easy_edm](#).

fastEDM 0.1 Reference Articles ▾ Changelog Search for

fastEDM



The `fastEDM` R package implements a series of *Empirical Dynamic Modeling* tools that can be used for *causal analysis of time series* data.

Key features of the package:

- powered by a fast multi-threaded C++ backend,
- able to process panel data, a.k.a. *multispatial EDM*,
- able to handle *missing data* using new `dt` algorithms or by dropping points.

Installation

You can install the development version of fastEDM from [GitHub](#) with:

License
[MIT + file LICENSE](#)

Citation
[Citing fastEDM](#)

Developers

Patrick Laub	Author, maintainer
Jinjing Li	Author
Michael Zyphur	Author
More about authors...	

Python package

≡ fastEDM Python Package

 Search

fastEDM Python Package

Package Description

Empirical Dynamic Modeling (EDM) is a way to perform *causal analysis on time series data*. The `fastEDM` Python package implements a series of EDM tools, including the convergent cross-mapping algorithm.

Key features of the package:

- powered by a fast multi-threaded *C++ backend*,
- able to process panel data, a.k.a. *multipatial EDM*,
- able to handle *missing data* using new `dt` algorithms or by dropping points.

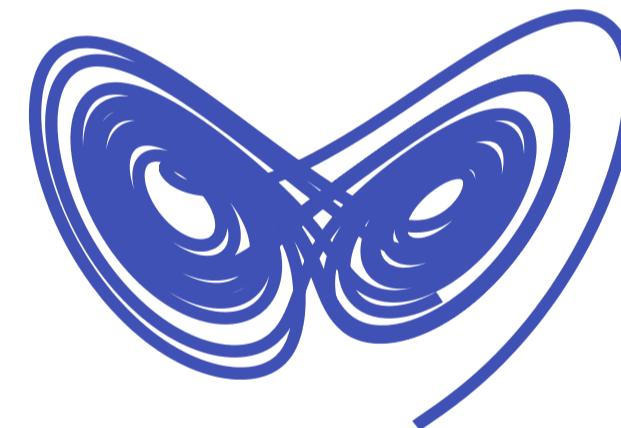


Table of contents

[Package Description](#)

[Installation](#)

[Example: Chicago crime levels and temperature](#)

[Stata & R packages](#)

[Other Resources](#)

[Authors](#)

[Citation](#)

Installation

To install the latest version from [Github](#) using `pip` run:

Modern engineering

- Open code (9,745 LOC) on MIT License,
- unit & integration tests (5,342 LOC),
- documentation (5,042 LOC),
- Git (1,198 commits),
- Github Actions (11 tasks),
- vectorised, microbenchmarking, ASAN, linting,
- all C++ compilers, WASM, all OSs.

Get involved!



Give it a try, feedback would be very welcome.



If you're talented in causal inference or programming (Stata/Mata, R, Javascript, C++, Python), we'd love contributions!