Reconstructing and predicting stochastic dynamical systems using probabilistic deep learning

Journal Club

Will An

Introduction

- This paper builds on the idea of taken’s theorem where states that a limited observation of a dynamical system can be used to reconstruct the full phase space.

- With the help of deep learning, it extracts the **stochastic features** while preserving the underlying dynamical structure.

- By utilizing temporal convolutional network (TCN), it can predict based on long-range temporal dependencies.

Introduction – Taken's Theorem

- Limited observations of state variables can retain the Lyapunov exponent by proper lag-embedding.

Given a time series γ:

$$\gamma = \{x_i\}, i = 1, \ldots, N, \tag{12}$$

we could recontruct the phase space $\mathbf{\Gamma}$ as :

$$\mathbf{\Gamma}^{m,\tau} = \left\{ \boldsymbol{X}_i^{m,\tau} = \left(x_i(\boldsymbol{k}), x_{i+\tau}(\boldsymbol{k}), \ldots, x_{i+(m-1)\tau}, (\boldsymbol{k}) \right) \right\}, i = 1, \ldots, N - m\tau + \tau. \tag{13}$$

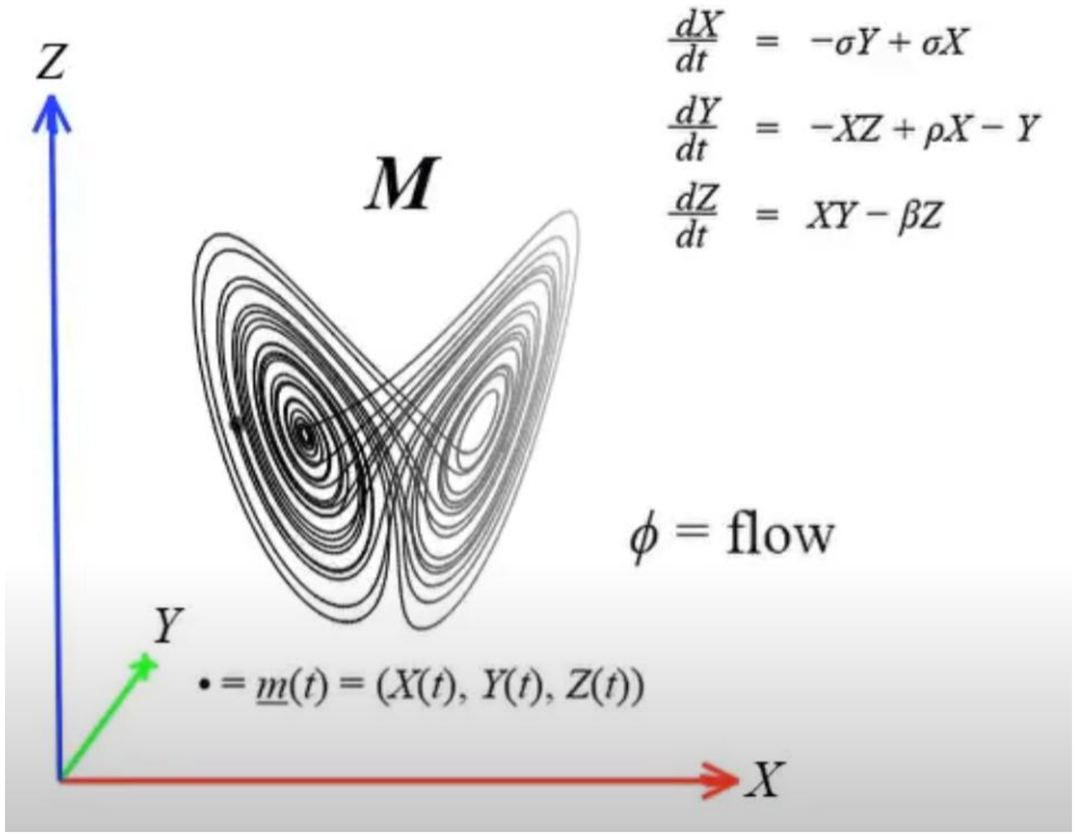
where

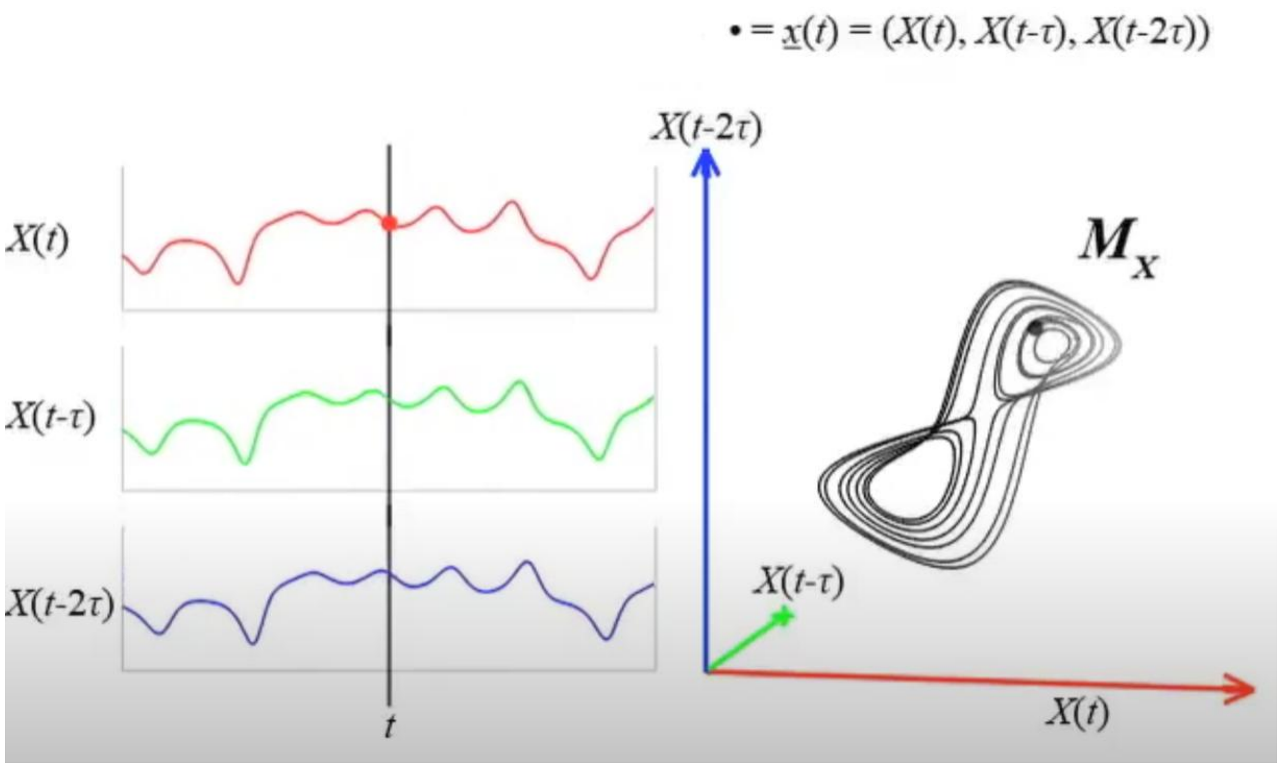
- i is the time step.
- m is the embedded dimension of the time series.
- τ is the lagging of the time series.
- N is the total time steps of the time series.

```
m = 2 # embedded dimension
tau = 2 # lagging

time_series = [x0, x1, x2, x3, x4, x5, x6]
time_series_embedded = [[x0, x2], [x1, x3], [x2, x4], [x3, x5], [x4, x6]]
```

Introduction – Taken's Theorem



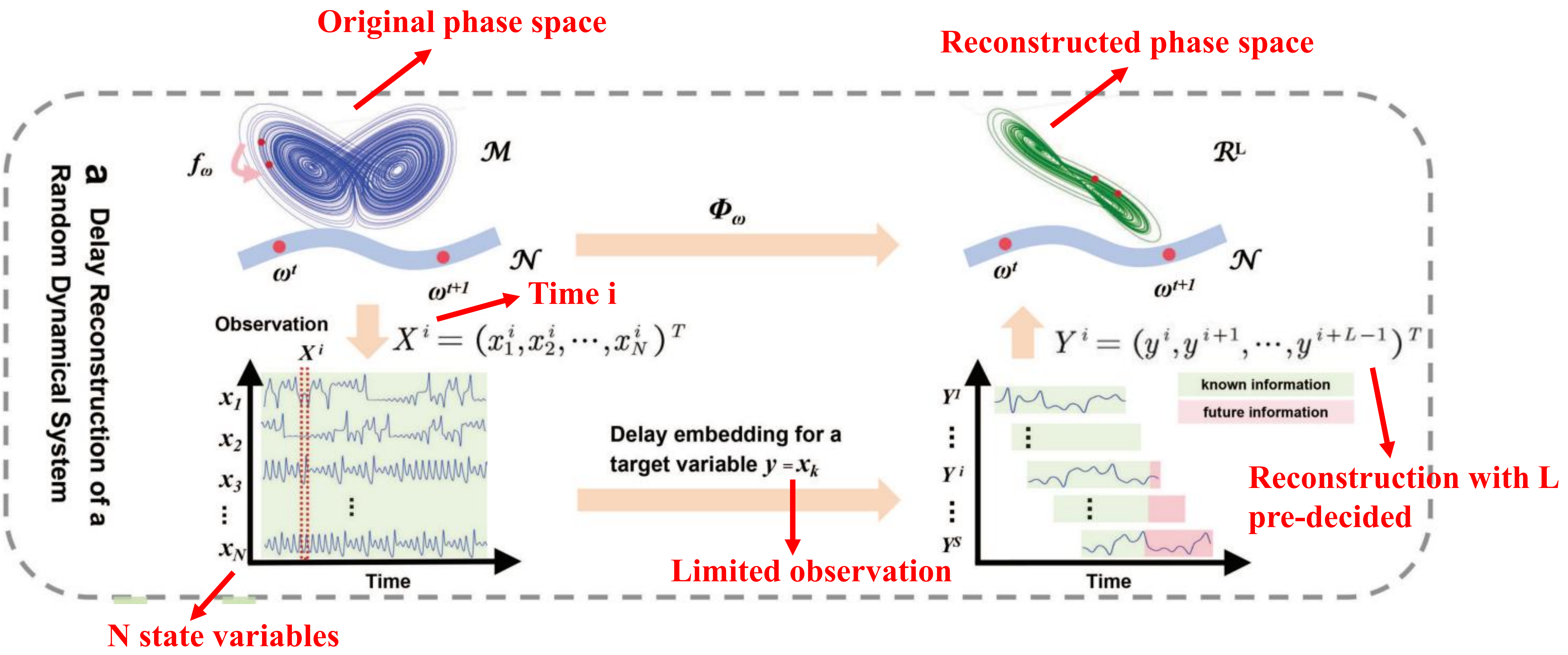


https://www.youtube.com/watch?v=6i57udsPKms&t=40s
Sugihara, George, et al. "Detecting causality in complex ecosystems." *science* 338.6106 (2012): 496-500.

Introduction – Stochastic dynamical system

- Normal case: $X^{i+1} = f(X^i)$ (X^i means all state variables at time i)
- Stochastic case: $X^{i+1} = f(X^i, \omega^i)$, (ω is white noise)

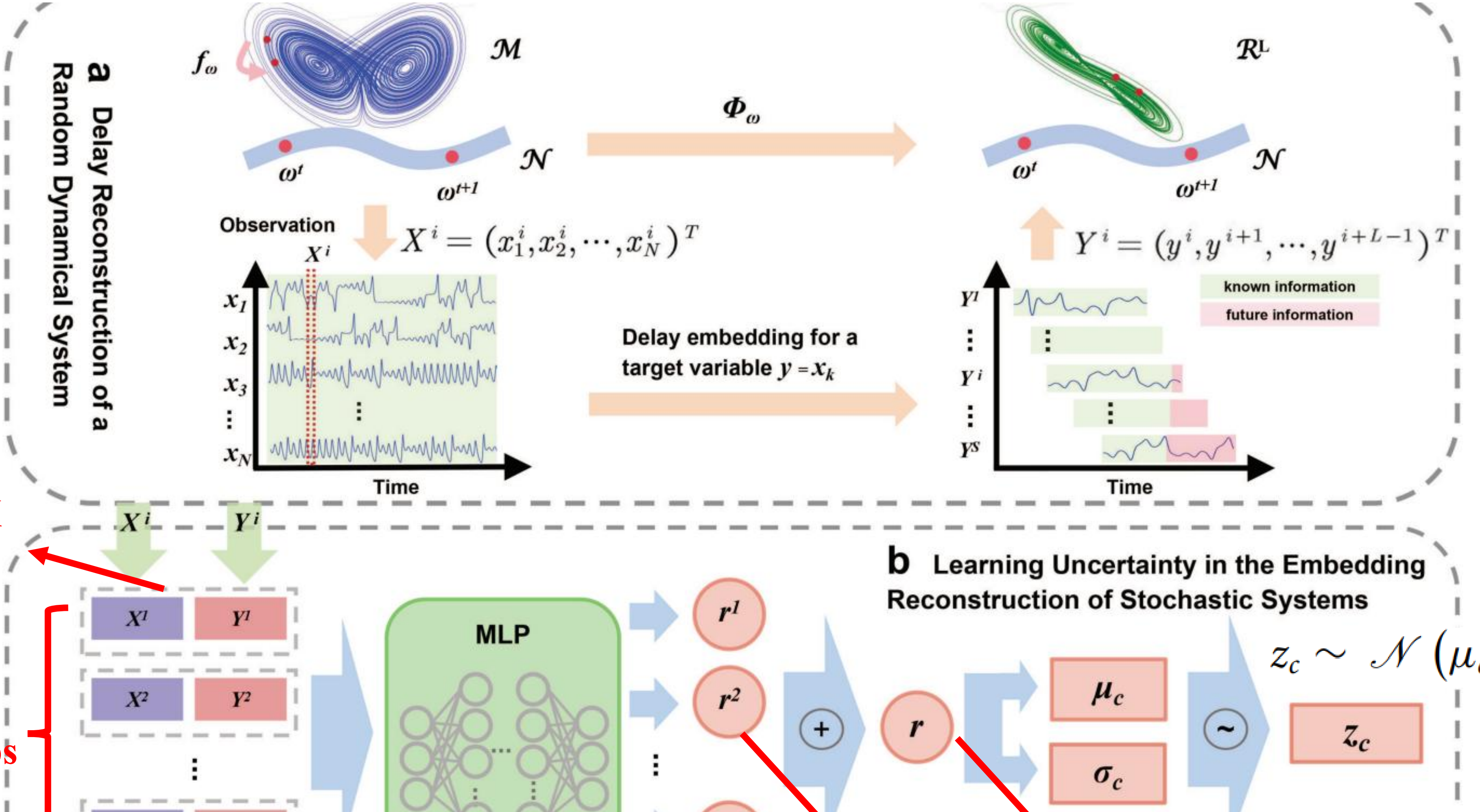
Method – lag-embedding



Method – Extract Stochastic Feature

Concatenate X and Y

Train based on first k time steps



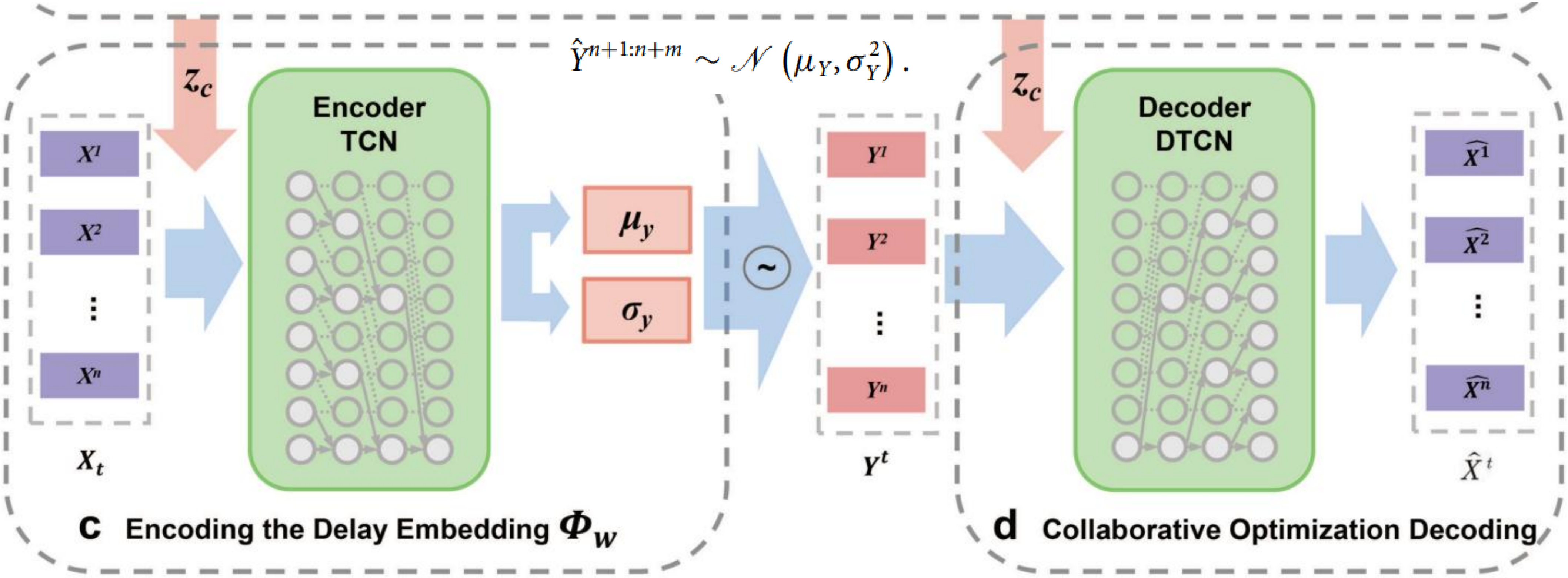
$$z_c \sim \mathcal{N}(\mu_c, \sigma_c^2)$$

$$r^i = h(X^i, Y^i)$$

$$(\mu_c, \sigma_c) = \frac{1}{k}\sum_{i=1}^{k} h(X^i, Y^i)$$

$$Loss_{KL} = \log \frac{q(z|X^{1:k}, Y^{1:k})}{q(z|X^{1:n}, Y^{1:n})}.$$

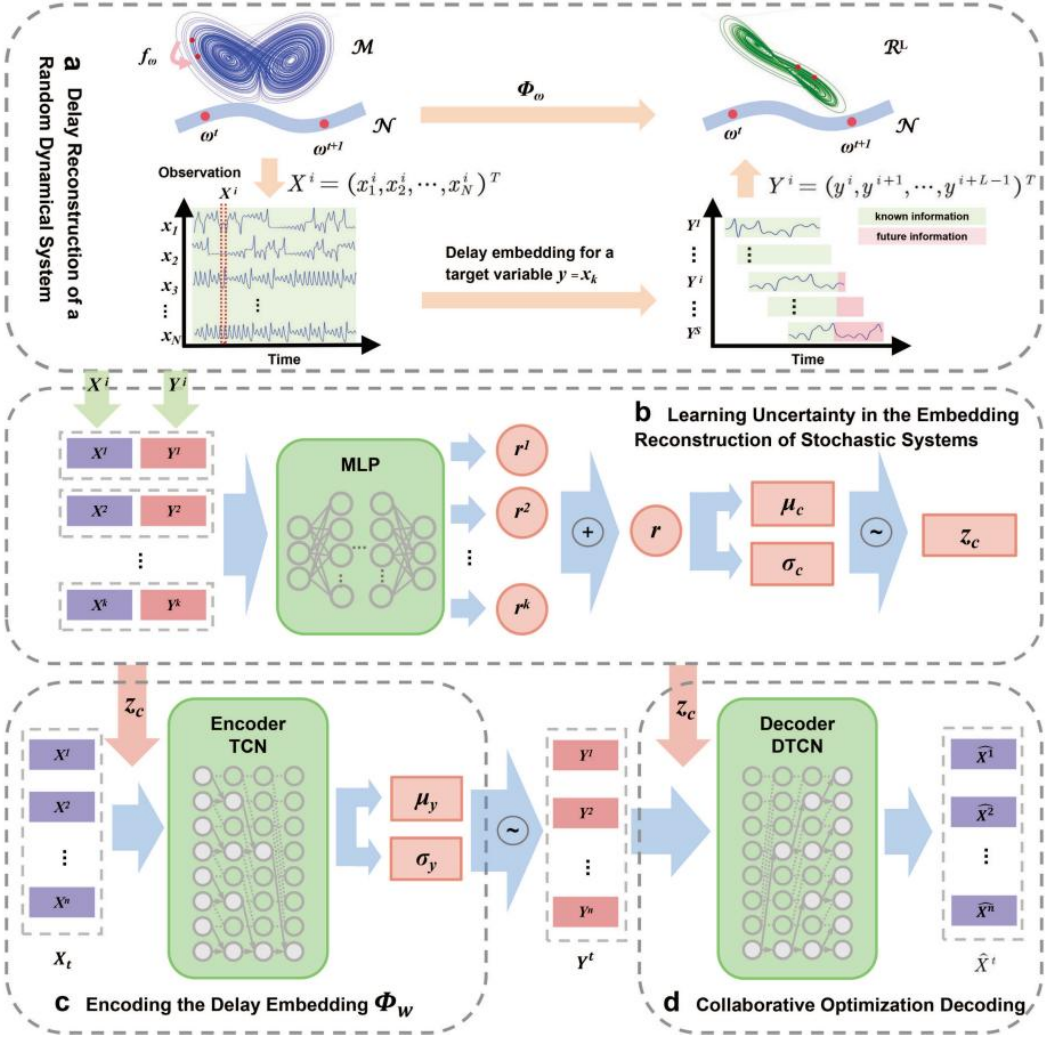
Method – TCN with stochasticity info



$$Loss_Y = -\sum_{i=n+1}^{n+m} \log p\left(\hat{Y}^i | z, X^i\right).$$

$$Loss_{Rec} = \sum_{i=n+1}^{n+m} \|X^i - \hat{X}^i\|_F.$$

Method – Model Training



$$Loss_{KL} = \log \frac{q\left(z|X^{1:k}, Y^{1:k}\right)}{q\left(z|X^{1:n}, Y^{1:n}\right)}.$$

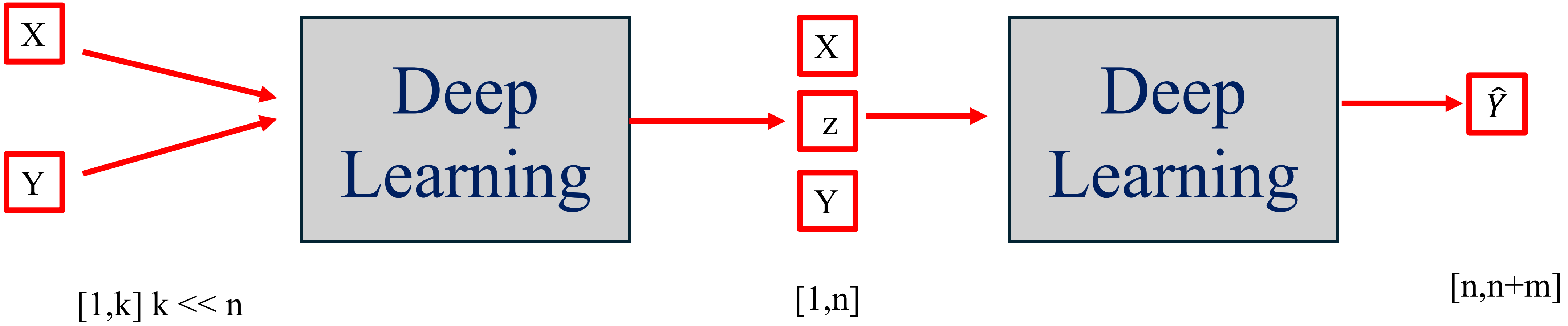
$$Loss_{Y} = -\sum_{i=n+1}^{n+m} \log p\left(\hat{Y}^{i}|z, X^{i}\right).$$

$$Loss_{Rec} = \sum_{i=n+1}^{n+m} \|X^{i} - \hat{X}^{i}\|_{F}.$$

$$Loss = \lambda_1 Loss_{Y} + \lambda_2 Loss_{Rec} + \lambda_3 Loss_{KDL}.$$

Method - Summary

- For a dynamical system, with time range [1,n], full state space X, target variable Y



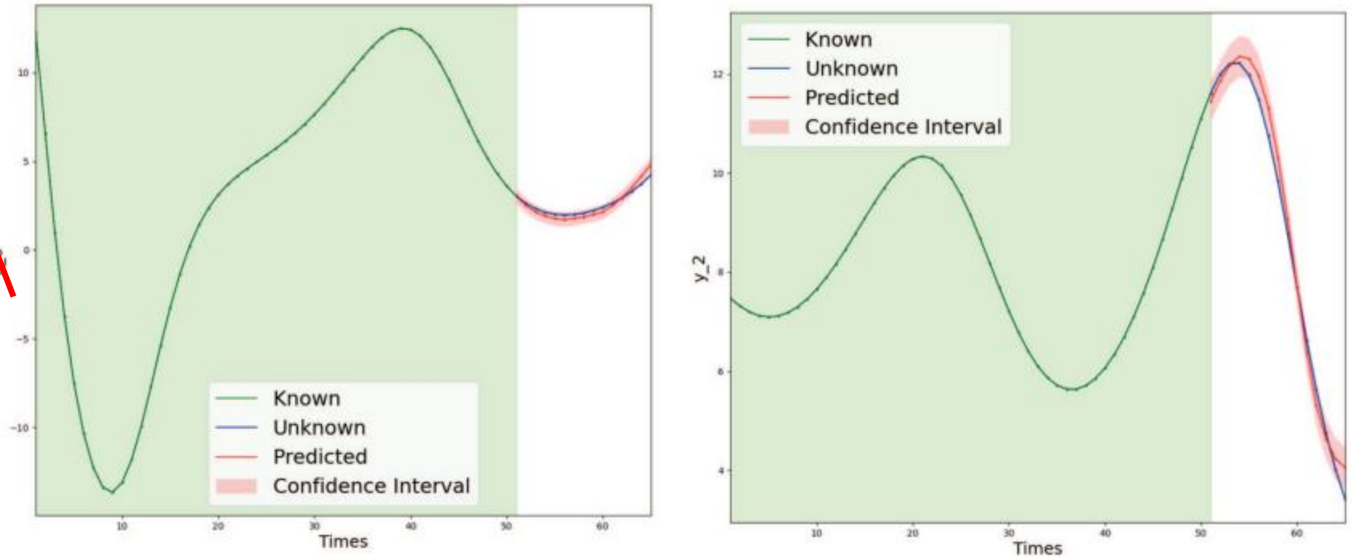
Results – Coupled Lorentz System

$$\begin{cases} \dot{x}_i = \sigma\left(y_i - x_i\right) + c x_{i-1}, \\ \dot{y}_i = \rho x_i - y_i - x_i z_i, \\ \dot{z}_i = x_i y_i - \beta z_i, \end{cases}$$

- i = 1,2,…,30
- Dimension = 90

where $c = 0.1,\ \rho = 28,\ \beta = \frac{8}{3},\ \sigma = 10.$

y_1

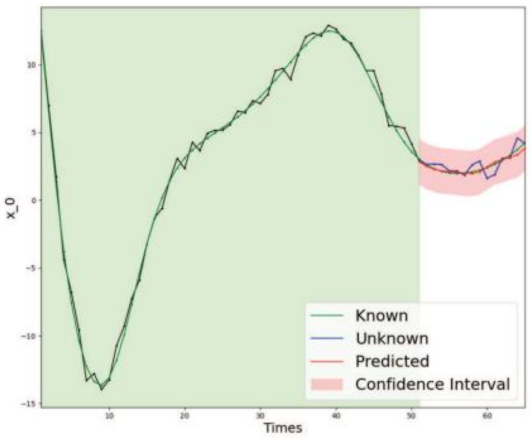


(a) (d)

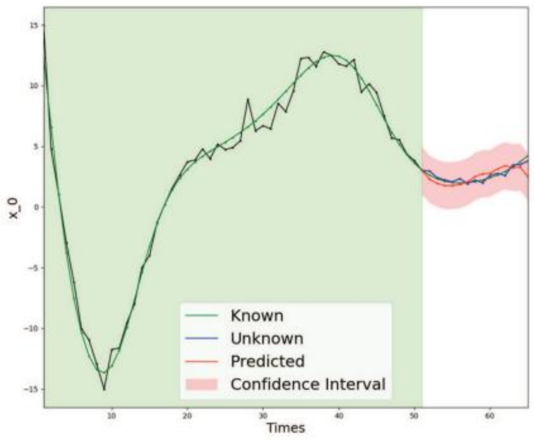
Noise free

. Two target variables, y_1 and y_2, were randomly selected from $\{x_1, x_2, \ldots, x_{90}\}$.

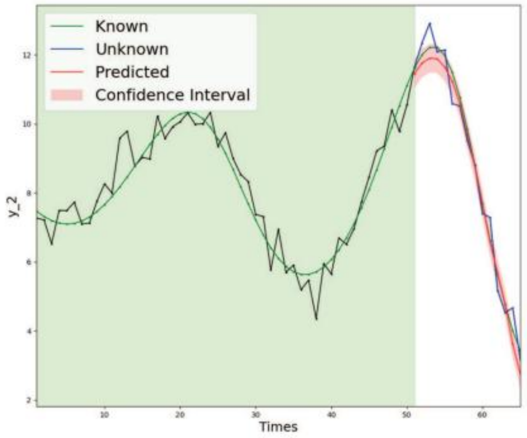
Results – Coupled Lorentz System



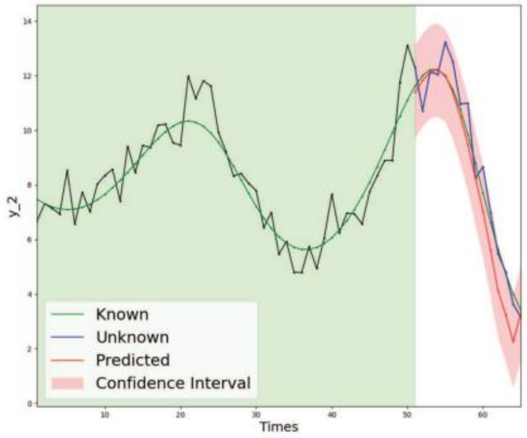
(b)



(c)



(e)



(f)

b,e: σ = 0.5

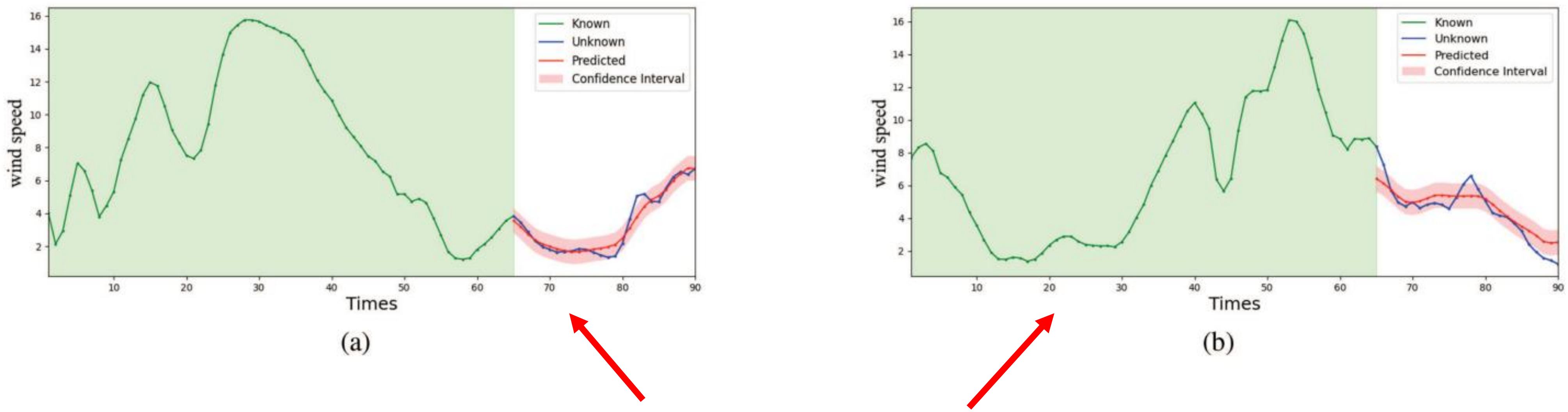
c,f: multiplicative white noise (μ = 1, σ = 0.1)

$$x_{\text{noisy}}(t) = x(t) \cdot \eta(t)$$

$$\eta(t) \sim \mathcal{N}(\mu = 1, \sigma = 0.1)$$

Results – Wind Speed

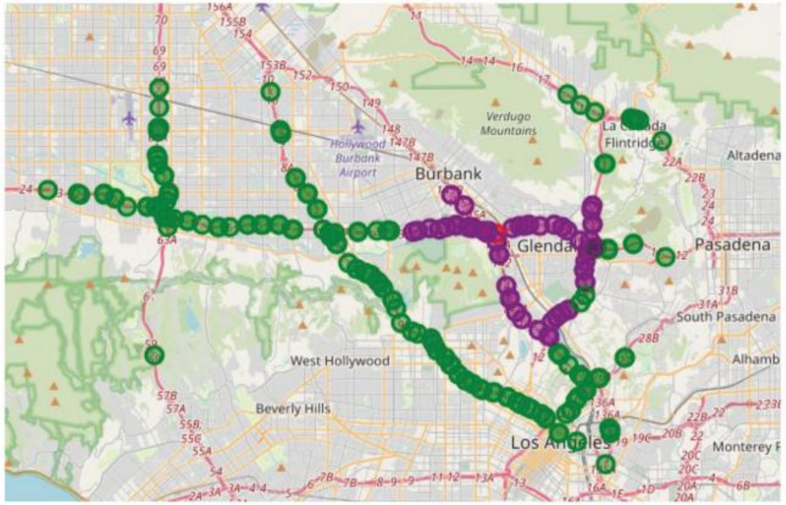
- Measured every 10 min at 155 wind stations across Japan.
- So it assumes full state space = 155 wind stations (X)
- It randomly selects a target station to predict (Y)



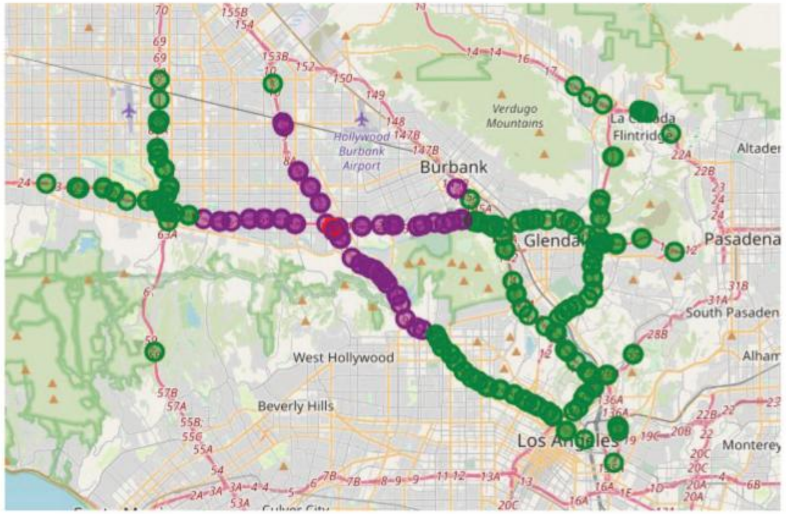
Two different periods, same station (Y)

Results – Traffic Speed

- 207 loop detectors along Highway 134 in California, USA, recorded every 5 min. 55 nearest-neighbor detectors were selected for each target sensor.
- So Y is the red dot. X are 55 purple nearby points.

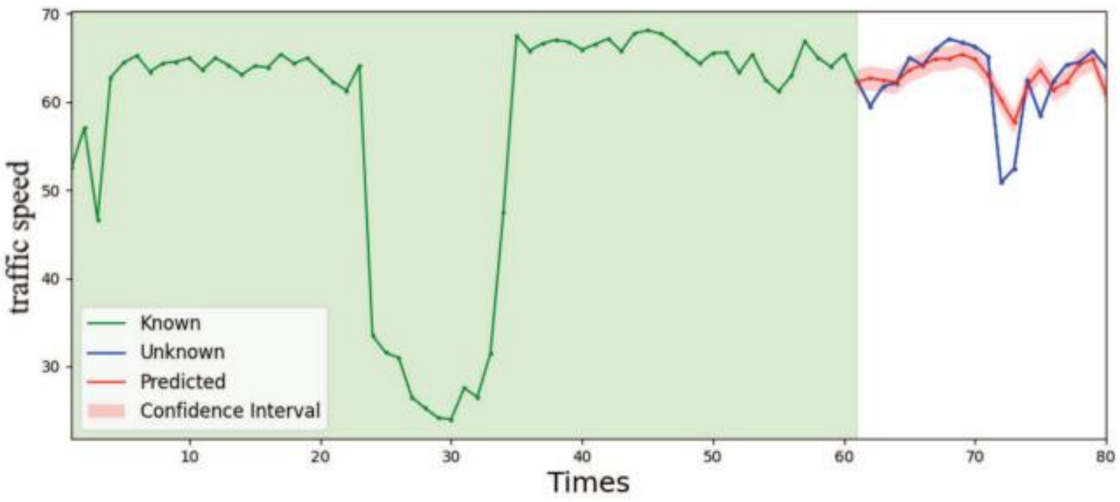


(c)

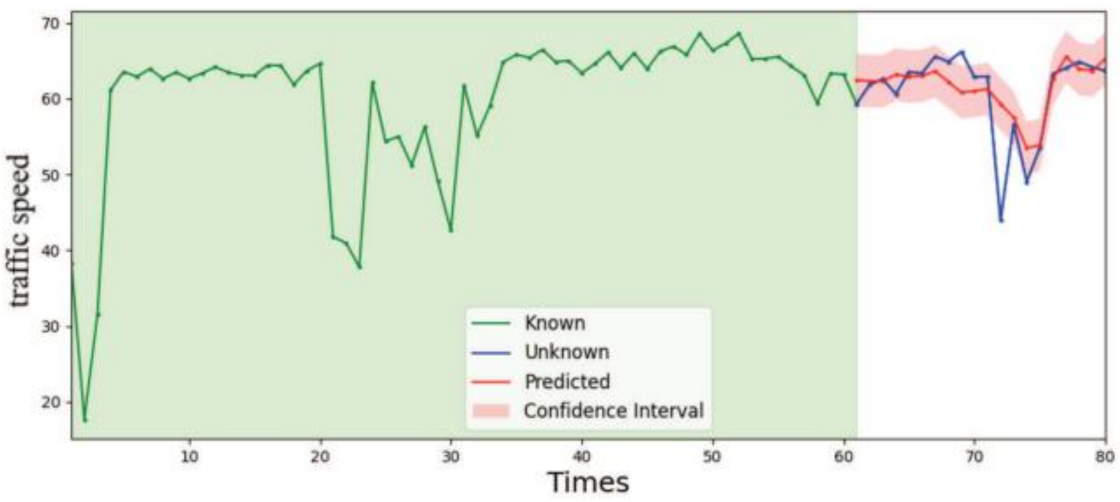


(d)

Results – Traffic Speed



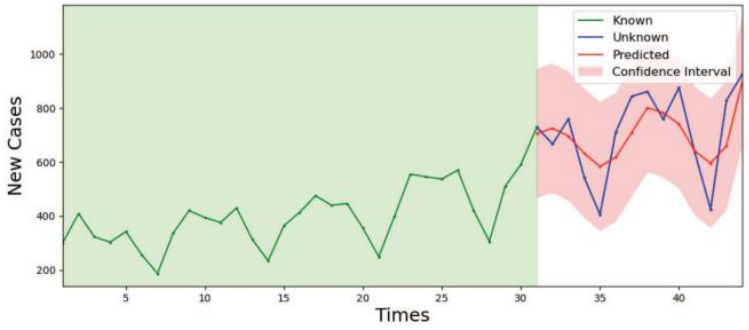
(a)



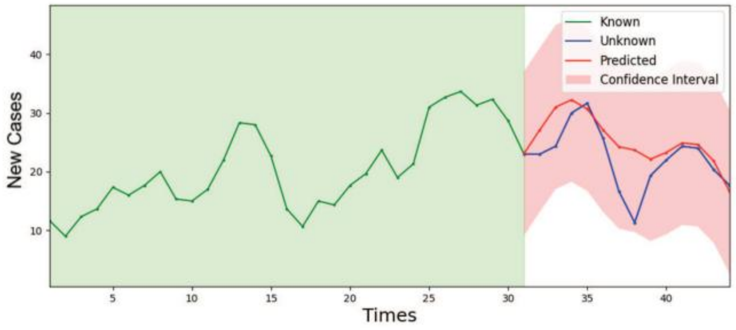
(b)

Results – Covid 19

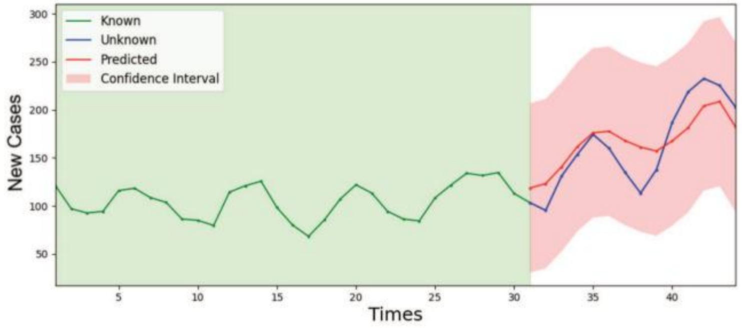
- Based on data from all 47 districts in Japan, the forecasts of daily new COVID-19 cases for four target cities are shown in Fig. 6(a) (Tokyo), Fig. 6(b) (Tochigi), Fig. 6(c) (Kanagawa), and Fig. 6(d) (Gunma).



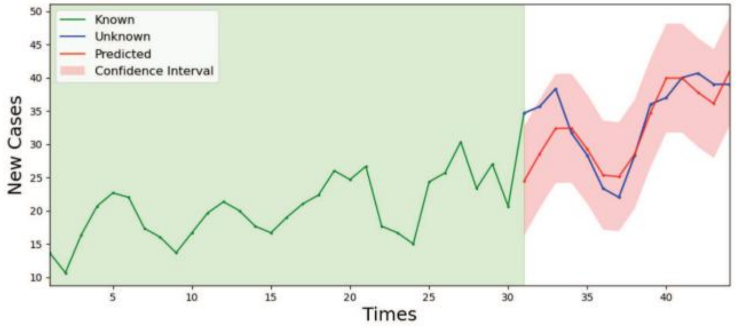
(a)



(b)



(c)



(d)

Results - Overall

TABLE I. Comparison of the performance among other forecasting methods.

	DSTDEM		ARIMA		VAR		LSTM		RBFN		SVR		STICM	
Datasets	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC	RMSE	PCC
Lorenz (noise-free)	0.037	0.997	0.434	0.878	0.320	0.820	0.137	0.946	1.798	−0.419	0.428	0.941	0.111	0.997
Lorenz (noise)	0.027	0.990	0.482	0.959	0.297	0.802	0.144	0.944	1.860	0.290	0.387	0.940	0.307	0.989
Wind speed	0.228	0.964	−1.24	0.595	0.452	−0.792	0.360	0.788	1.985	0.873	0.494	0.136	0.908	0.942
Traffic speed	0.548	0.959	0.639	−0.499	0.383	0.501	0.314	0.568	6.616	−0.181	0.449	0.181	0.66	0.901
COVID-19	0.219	0.935	0.799	−0.220	0.485	0.739	0.660	−0.022	3.480	0.405	0.852	0.033	0.608	0.9

$$\mathrm{PCC}(y,\hat{y}) = \frac{\mathrm{Cov}(y,\hat{y})}{\sigma_y \sigma_{\hat{y}}} = \frac{\sum_t (y_t - \bar{y})(\hat{y}_t - \bar{\hat{y}})}{\sqrt{\sum_t (y_t - \bar{y})^2}\sqrt{\sum_t (\hat{y}_t - \bar{\hat{y}})^2}}$$

Conclusion

- The DSTDEM is grounded in the embedding reconstruction theory and incorporates an advanced deep probabilistic framework, offering a novel perspective for analyzing high-dimensional time-series data.
- The DSTDEM effectively captures features across different time scales in time series using the temporal convolution structure.
- Also, it effectively extracts stochastic features from historical time-series data.
- DSTDEM has been validated through applications in various real-world scenarios