Supplementary Material

1 Supplementary Note 1

1.1 Autoencoder

Autoencoder (Wang et al. 2014) is a class of neural network that can compress high-dimensional state into low-dimensional form by encoder χ_e and restore low-dimensional state to high-dimensional form by decoder χ_d.

Here, Eq. 1 shows how to map the high-dimensional inputting state $\boldsymbol{F^t}$ onto reconstructed high-dimensional state $\widehat{\boldsymbol{F}}^{\boldsymbol{t}}$,

$$\widehat{\boldsymbol{F}}^{\boldsymbol{t}} = \chi_d \circ \chi_e(\boldsymbol{F^t}) \tag{1}$$

where $\circ$ is the function composition operation.

Previously, Lusch et al. (Lusch, Kutz and Brunton 2018) build a deep auto-encoder framework to accurately predict the future state of metabolomics time series with flow behavior, which demonstrates that autoencoder is good at processing nonlinear systems.

1.2 Delay embedding theory

For a high dimensional nonlinear system with n-dimensional variables, we define $\boldsymbol{F^t} = (f_1^t, f_2^t, \dots, f_n^t)'$ as the observed non-delay attractor, which represents the state of the system at time step t in the n-dimensional space. Here “'” is the transpose of a vector. The delay embedding theory (Sauer, Yorke and Casdagli 1991, Holmes et al. 2012) suggests that, the mapping $\Phi: \mathbb{R}^n \to \mathbb{R}^L$ to the observed non-delay attractor $\boldsymbol{F^t}$ is an embedding when $L > 2d$ (d denotes the box-counting dimension of $\boldsymbol{F^t}$), and a delay attractor $\boldsymbol{Y^t} = (y^t, y^{t+1}, \dots, y^{t+L-1})'$ of length L can be constructed by Eq. 2.

$$\Phi(\boldsymbol{F^t}) = (y^t, y^{t+1}, \dots, y^{t+L-1})' = \boldsymbol{Y^t} \tag{2}$$

Moreover, the mapping between $\boldsymbol{F^t}$ and $\boldsymbol{Y^t}$ is a one-to-one map with the conjugate form, Chen et al. (Chen et al. 2020) has derived the conjugate form of Φ as $\Psi: \mathbb{R}^L \to \mathbb{R}^n$ (Eq. 3).

$$\Psi(\boldsymbol{Y^t}) = (f_1^t, f_2^t, \dots, f_n^t)' = \boldsymbol{F^t} \tag{3}$$

Previous studies have predicted the future state of multi-omics time series based on the delay embedding theory (Sauer et al. 1991, Holmes et al. 2012). For example, Chen et al. (Chen et al. 2020) proposed an Anticipated Learning Machine (ALM) to achieve precise future-state prediction of time series with chaotic behavior related to genomics.

1.3 Koopman theory

Koopman theory (Koopman 1931) provides a new research direction in terms of dealing with complex nonlinear relations, which suggests that a nonlinear dynamical system can undergo a transformation into an infinite dimensional space, in which it evolves linearly in time by virtue of Koopman operator (Mezić 2005, Budišić, Mohr and Mezić 2012).

However, we need to find an approximate finite dimensional representation for the infinite dimensional Koopman operator in practice. Rice et al. (Rice, Xu and August 2020) assumed that there exists a mapping that can approximate Koopman operator to a finite dimension linear matrix, which can learn the forward (or backward) dynamics of system. Based on Koopman theory, Azencot et al. (Azencot et al. 2020) develop a Physics Constrained Learning framework to accurately predict the future state of proteomics time series data with oscillating behavior metabolomics.

2 Supplementary Note 2

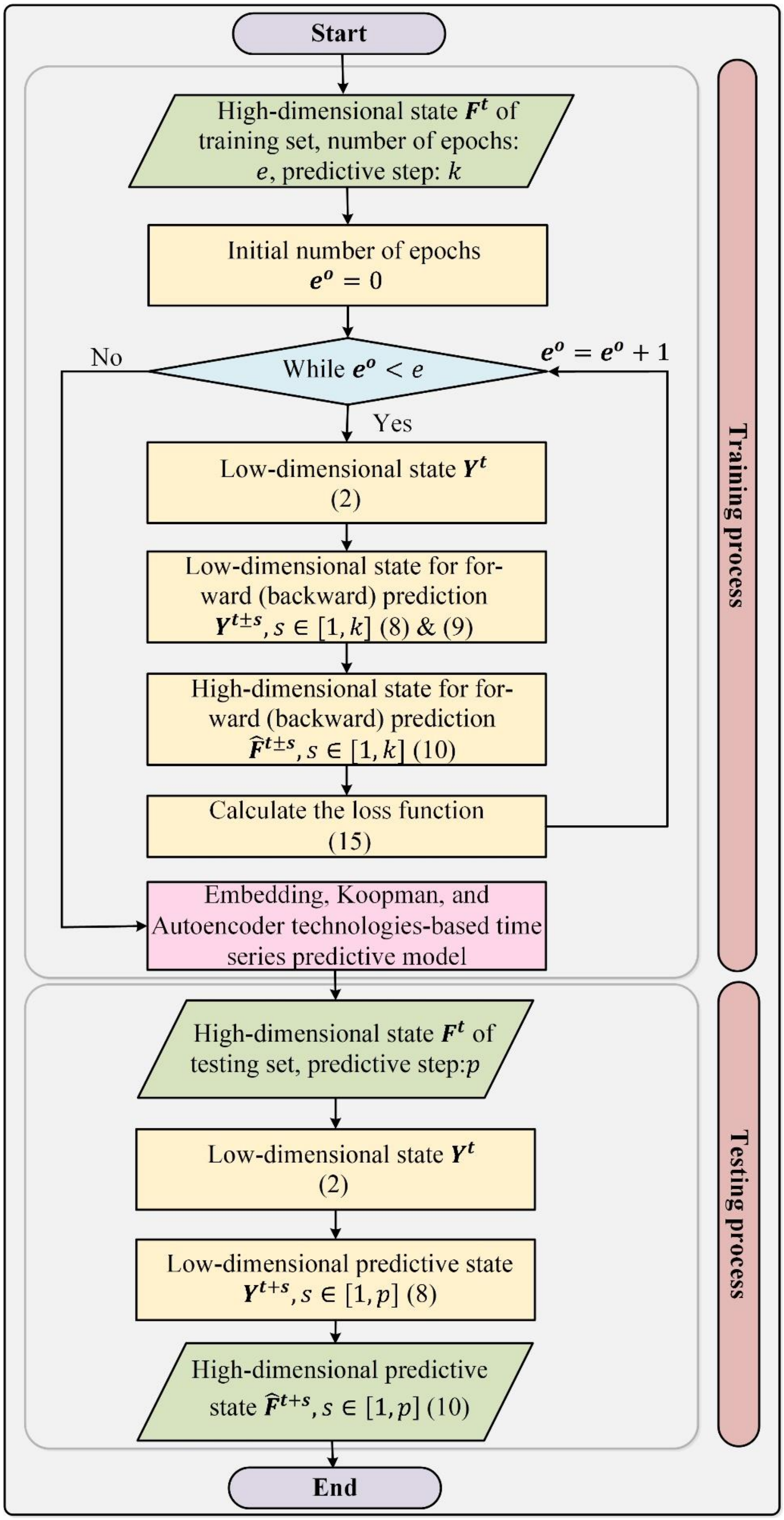
We set the experimental environment as follows: we use deep learning framework PyTorch to implement the EKATP. The experimental hardware environment is configured by Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz, the memory of which is 16.0 GB.

Our experiment employs the following benchmark methods for predictive performance comparison, which are recurrent neural network (RNN) (Jiang and Lai 2019), long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), dynamic Autoencoder (DAE) (Lusch et al. 2018) and Koopman Autoencoder (KAE) (Azencot et al. 2020). We detail these methods as follows.

RNN (Jiang et al. 2019) is a classical deep neural network, usually used for sequential learning tasks; LSTM (Hochreiter et al. 1997) is the variant of the classical RNN, which introduces the concept of cell state to improve the structure of RNN; DAE (Lusch et al. 2018) develops the deep neural network representations of Koopman eigenfunctions, which is often used for high-dimensional and nonlinear systems; KAE (Azencot et al. 2020) is a physically constrained learning model based on the Koopman theory for high-dimensional time series data processing.

3 Supplementary Figure 1

We describe the training process and testing process of EKATP in detail as shown in Supplementary Figure 1.


Start
High-dimensional state $\boldsymbol{F}^t$ of training set, number of epochs: e, predictive step: k
Initial number of epochs $\boldsymbol{e}^{\boldsymbol{o}} = 0$
No
While $\boldsymbol{e}^{\boldsymbol{o}} < e$
$\boldsymbol{e}^{\boldsymbol{o}} = \boldsymbol{e}^{\boldsymbol{o}} + 1$
Yes
Low-dimensional state $\boldsymbol{Y}^t$ (2)
Low-dimensional state for forward (backward) prediction $\boldsymbol{Y}^{t\pm s}, s \in [1,k]$ (8) & (9)
High-dimensional state for forward (backward) prediction $\widehat{\boldsymbol{F}}^{t\pm s}, s \in [1,k]$ (10)
Calculate the loss function (15)
Embedding, Koopman, and Autoencoder technologies-based time series predictive model
High-dimensional state $\boldsymbol{F}^t$ of testing set, predictive step:p
Low-dimensional state $\boldsymbol{Y}^t$ (2)
Low-dimensional predictive state $\boldsymbol{Y}^{t+s}, s \in [1,p]$ (8)
High-dimensional predictive state $\widehat{\boldsymbol{F}}^{t+s}, s \in [1,p]$ (10)
End
Training process
Testing process

Supplementary Figure 1 The calculation flow chart of EKATP

4 Supplementary Method 1

4.1 Pseudo code of training process

```
Input: High-dimensional nonlinear time series state $\boldsymbol{F}^{\boldsymbol{t}} = (f_1^t, f_2^t, \dots f_n^t)'$ of training set, number of epochs: $e$, predictive step: $k$
For $e^0 = 0, \dots, e$
    $\boldsymbol{Y}^{\boldsymbol{t}} \leftarrow \chi_e(\boldsymbol{F}^{\boldsymbol{t}})$    obtain low-dimensional state at time $t$ by (2)
    $\boldsymbol{Y}^{\boldsymbol{t}+s} \leftarrow C^s \boldsymbol{Y}^{\boldsymbol{t}}, s \in [1, k]$    obtain low-dimensional state at time $t + s$ by (8)
    $\boldsymbol{Y}^{\boldsymbol{t}-s} \leftarrow D^s \boldsymbol{Y}^{\boldsymbol{t}}, s \in [1, k]$    obtain low-dimensional state at time $t - s$ by (9)
    $\widehat{\boldsymbol{F}}^{\boldsymbol{t} \pm \boldsymbol{s}} \leftarrow \chi_d(\boldsymbol{Y}^{\boldsymbol{t} \pm \boldsymbol{s}}), s \in [1, k]$    obtain high-dimensional state at time $t \pm s$ by (10)
    Calculate the loss function by (15)
End For
Output: trained EKATP
```

4.2 Pseudo code of testing process

```
Input: High-dimensional nonlinear time series state $\boldsymbol{F}^{\boldsymbol{t}} = (f_1^t, f_2^t, \dots f_n^t)'$ of testing set, predictive step: $p$, trained EKATP
$\boldsymbol{Y}^{\boldsymbol{t}} \leftarrow \chi_e(\boldsymbol{F}^{\boldsymbol{t}})$    obtain low-dimensional state at time $t$ by (2)
$\boldsymbol{Y}^{\boldsymbol{t}+s} \leftarrow C^s \boldsymbol{Y}^{\boldsymbol{t}}, s \in [1, p]$    obtain low-dimensional state at time $t + s$ by (8)
$\widehat{\boldsymbol{F}}^{\boldsymbol{t}+\boldsymbol{s}} \leftarrow \chi_d(\boldsymbol{Y}^{\boldsymbol{t}+\boldsymbol{s}}), s \in [1, p]$    obtain high-dimensional state at time $t + s$ by (10)
Output: High-dimensional predictive state $\widehat{\boldsymbol{F}}^{\boldsymbol{t}+\boldsymbol{s}}, s \in [1, p]$
```

5 Supplementary Method 2

To evaluate the experimental results, we use the relative predictive error, which is computed at each time step t via Eq. 4:

$$Predictive\ error = \left\|\boldsymbol{F}_{\boldsymbol{t}} - \widehat{\boldsymbol{F}}_{\boldsymbol{t}}\right\|_2 / \|\boldsymbol{F}_{\boldsymbol{t}}\|_2 \tag{4}$$

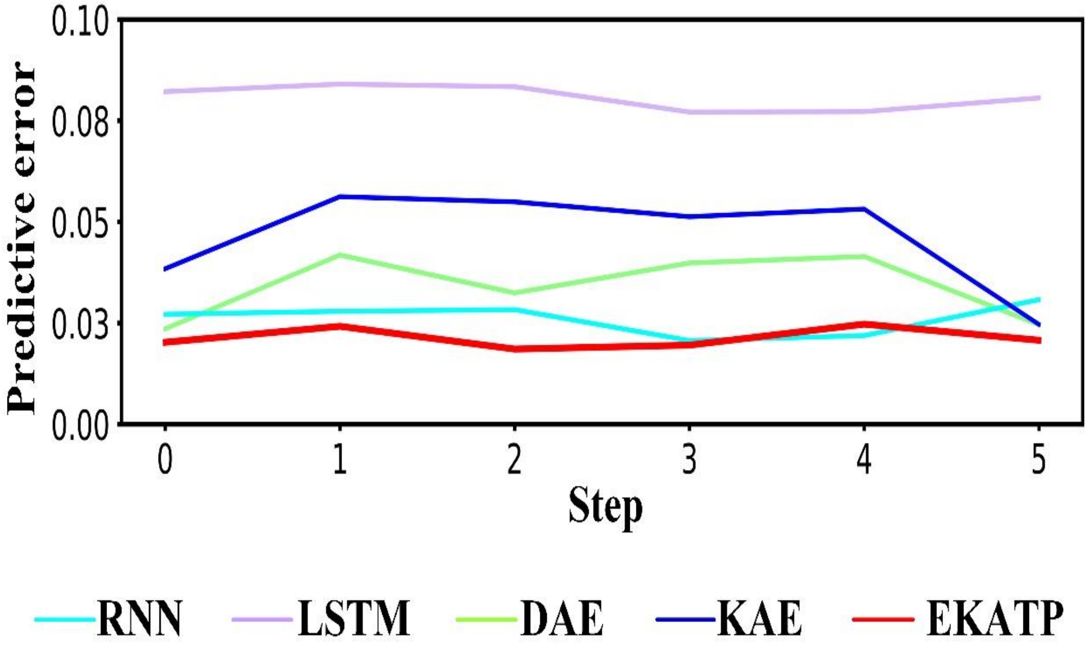
where $\boldsymbol{F}_{\boldsymbol{t}}$ represents true state, $\widehat{\boldsymbol{F}}_{\boldsymbol{t}}$ represents predictive state. In EKATP model and KAE model (Azencot et al. 2020), we average the errors over 30 different initial observations, where the shaded areas represent the ± 1 standard deviations.

6 Supplementary Experiment

In order to prove the performance of EKATP on real datasets, we use the gene expression data of circadian rhythm (Wang, Zhang and Chen 2009) to test the performance of EKATP. Also, we listed the detailed data in Supplementary Table 4.1. The gene expression data contains 84 dimensions and T=21 steps. Here, we employ last six steps to visualize the predictability for the EKATP.

Supplementary Figure 2 shows the the predictive error for the EKATP and other existing methods, the details of which are listed in Supplementary Table 4.2 and Supplementary Method.

Supplementary Figure 2 indicates that since EKATP has less of a predictive error than the existing methods, we consider that EKATP outperforms the other existing methods.



Supplementary Figure 2 Comparison among the RNN, LSTM, DAE, KAE and EKATP. The abscissa represents the step, and the ordinate represents the predictive error.

7 Supplementary Table 1

Genomics dataset

Table 1.1 The data of low-dimensional gene expression time series $\{v_t\}$ under the condition of h=0.003 and T=1050

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 1.1.csv)

Table 1.2 The data of low-dimensional gene expression time series $\{v_t\}$ under the condition of h=0.006 and T=1050

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 1.2.csv)

Table 1.3 The data of low-dimensional gene expression time series with T=15000

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 1.3.csv)

Table 1.4 The data of high-dimensional gene expression time series $\{f_t\}$ under the condition of h=0.003 and T=1050

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 1.4.csv)

Table 1.5 The data of high-dimensional gene expression time series $\{f_t\}$ under the condition of h=0.006 and T=1050

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 1.5.csv)

Table 1.6 The data of high-dimensional gene expression time series with T=15000.

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 1.6.csv)

Table 1.7 Parameter setting of Lorenz system

Parameter	η	ρ	β
value	10.0	28.0	8.0/3.0

Table 1.8 Division of training set and testing set

Datasets	Training set	Testing set
Value	[0:1000]	[1000:1050]

Table 1.9 The interval of gene time series in three different periods

Parameter	T	t_1	t_2	t_3
value	15000	[2000:3000]	[2700:3700]	[5780:6580]

8 Supplementary Table 2

Proteomics dataset

Table 2.1 The data of low-dimensional protein time series $\{v_t\}$ under the condition of h=0.8 and T=1600

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 2.1.csv)

Table 2.2 The data of low-dimensional protein time series $\{v_t\}$ under the condition of h=2.4 and T=1600

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 2.2.csv)

Table 2.3 The data of high-dimensional protein time series $\{v_t\}$ under the condition of h=0.8 and T=1600

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 2.3.csv)

Table 2.4 The data of high-dimensional protein time series $\{v_t\}$ under the condition of h=2.4 and T=1600

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 2.4.csv)

Table 2.5 Parameter setting of pendulum system

Parameter	l	g
value	1	9.8

Table 2.6 Division of training set and testing set

Datasets	Training set	Testing set
Value	[0:600]	[600:1600]

Table 2.7 The seed value of each condition

Condition	h=0.8 σ=0.00	h=0.8 σ=0.03	h=2.4 σ=0.00	h=2.4 σ=0.00
Seed value	1-20	21-40	121-140	141-160

9 Supplementary Table 3

Metabolomics dataset

Table 3.1 The data of low-dimensional metabolic time series $\{v_t\}$ under the condition of ζ_1 and T=900 (Lusch et al. 2018)

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 3.1.csv)

Table 3.2 The data of low-dimensional metabolic time series $\{v_t\}$ under the condition of ζ_2 and T=900 (Lusch et al. 2018)

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 3.2.csv)

Table 3.3 The data of high-dimensional metabolic time series $\{v_t\}$ under the condition of ζ_1 and T=900

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 3.3.csv)

Table 3.4 The data of high-dimensional metabolic time series $\{v_t\}$ under the condition of ζ_2 and T=900

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 3.4.csv)

Table 3.5 Parameter setting of fluid flow system

Parameter	γ	ω	A	λ
Value	0.1	1	-0.1	10

Table 3.6 Division of training set and testing set

Dataset	Training set	Testing set

Value	[0:800]	[800:900]

Table 3.7 PCC and RMSE values between predictive state and true state under different noise intensities

σ	0.001	0.005	0.010	0.050	0.100	0.500
PCC	0.999818	0.999726	0.999418	0.945674	0.768908	0.166338
RMSE	0.000225	0.000254	0.000332	0.003008	0.006405	0.015410

10 Supplementary Table 4

Table 4.1 The gene expression data of circadian rhythm

The data is listed on https://github.com/suranl/EKATP (Supplementary Table 4.1.csv)

Table 4.2 Division of training set and testing set

Dataset	Training set	Testing set
Value	[0:15]	[15:21]

P.S. Our code is available at https://github.com/suranl/EKATP

References:

Azencot, O., N. B. Erichson, V. Lin and M. Mahoney (2020). Forecasting sequential data using consistent Koopman autoencoders. International Conference on Machine Learning.

Budišić, M., R. Mohr and I. Mezić (2012). "Applied koopmanism." Chaos: An Interdisciplinary Journal of Nonlinear Science **22**(4): 047510.

Chen, C., R. Li, L. Shu, Z. He, J. Wang, C. Zhang, H. Ma, K. Aihara and L. Chen (2020). "Predicting future dynamics from short-term time series using an Anticipated Learning Machine." National Science Review **7**(6): 1079-1091.

Chen, P., R. Liu, K. Aihara and L. Chen (2020). "Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation." Nature communications **11**(1): 1-15.

Hochreiter, S. and J. Schmidhuber (1997). "Long short-term memory." Neural computation **9**(8): 1735-1780.

Holmes, P., J. L. Lumley, G. Berkooz and C. W. Rowley (2012). Turbulence, coherent structures, dynamical systems and symmetry, Cambridge university press.

Jiang, J. and Y.-C. Lai (2019). "Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius." Physical Review Research **1**(3): 033056.DOI: 10.1103/PhysRevResearch.1.033056.

Koopman, B. O. (1931). "Hamiltonian systems and transformation in Hilbert space." Proceedings of the national academy of sciences of the united states of america **17**(5): 315.

Lusch, B., J. N. Kutz and S. L. Brunton (2018). "Deep learning for universal linear embeddings of nonlinear dynamics." Nature communications **9**(1): 1-10.

Mezić, I. (2005). "Spectral properties of dynamical systems, model reduction and decompositions." Nonlinear Dynamics **41**(1): 309-325.

Rice, J., W. Xu and A. August (2020). "Analyzing Koopman approaches to physics-informed machine learning for long-term sea-surface temperature forecasting." arXiv preprint arXiv:2010.00399.

Sauer, T., J. A. Yorke and M. Casdagli (1991). "Embedology." Journal of statistical Physics **65**(3): 579-616.

Wang, W., Y. Huang, Y. Wang and L. Wang (2014). Generalized autoencoder: A neural network framework for dimensionality reduction. Proceedings of the IEEE conference on computer vision and pattern recognition workshops.

Wang, Y., X.-S. Zhang and L. Chen (2009). "A network biology study on circadian rhythm by integrating various omics data." OMICS A Journal of Integrative Biology **13**(4): 313-324.