Supplementary Materials

Predicting Future State of Nonlinear Dynamical Systems with Multi-View Embedding

Zheng Jiang[a,b] Yifang Ma[c] Wei Chen[a,b,d,*] Jürgen Kurths[e,f,*]

[a] School of Artificial Intelligence, Beihang University, Beijing, 100191, China

[b] Key Laboratory of Mathematics, Informatics and Behavioral Semantics (LMIB), Beihang University, Beijing, 100191, China

[c] Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, China

[d] Zhongguancun Laboratory, Beijing, 100191, China

[e] Potsdam Institute for Climate Impact Research(PIK) - Member of the Leibniz Association, Potsdam, 14473, Germany

[f] Department of Physics, Humboldt University at Berlin, Berlin, 12489, Germany

*E-mail: chwei@buaa.edu.cn, kurths@pik-potsdam.de

Contents

1. Constructing Delay Attractors and Non-delay Attractors by Embedding Theory

1.1 Construction of non-delayed attractors

For a general dynamical system with *n* state variables, the system state observed at *m* time points during its evolution at equal time intervals τ can be represented as a set of time series $\mathbf{X}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T, t = t_0, t_0 + \tau, \ldots, t_0 + (m-1)\tau$. After a sufficiently long period of time, the system will reach stability and all states of the system will converge to a compact manifold $\boldsymbol{\mathcal{V}}$. The box-counting dimension of the system attractor $\boldsymbol{\mathcal{A}}$ contained in the manifold $\boldsymbol{\mathcal{V}}$ is denoted as *d* (*d* is usually small). Whitney embedding theorem and generalized embedding theorem indicate that for any function h_k with at least second-order smoothness on $\boldsymbol{\mathcal{A}}$, the mapping $\boldsymbol{\Psi}_{<h_k>}: \boldsymbol{\mathcal{A}} \rightarrow \mathbb{R}^E$:

$$\boldsymbol{\Psi}_{<h_k>} = (h_1(\boldsymbol{X}), h_2(\boldsymbol{X}), \ldots, h_k(\boldsymbol{X}))$$

is generally an embedding when $E \geqslant 2d+1$. This means that as long as the embedding dimension *E* is sufficiently large, the dynamics of the original high-dimensional system can be reconstructed through the non-delayed embedding of the observed variables. The reconstructed non-delayed attractor is a diffeomorphism with the original attractor manifold, but does not contain time-lag information, so it cannot be directly used for predicting future states.

1.2 Construction of multi-view delay attractors

In order to synchronize the future dynamics of multiple state variables in a predictive system, we consider an embedding consisting of multiple system variables with different delays. Let ϕ represent a flow corresponding to an *n*-dimensional dynamical system. The generalized delay embedding theorem shows that under appropriate assumptions on the system, the mapping $\boldsymbol{\Psi}_{<h_k, \phi>} = (h_1(\boldsymbol{X}), h_2(\boldsymbol{X}), \ldots, h_k(\boldsymbol{X})), \boldsymbol{\mathcal{A}} \rightarrow \mathbb{R}^E$ driven by smooth functions h_k with delay is also generally an embedding. A function with delay refers to a function h_i that allows for a delay under ϕ, that is, it can be expressed as $h_i = \phi^b(h_j)$. Therefore, by selecting an appropriate embedding dimension, we can obtain a differential homeomorphism from the original attractor manifold to a delay embedding space consisting of multiple variables and their lags. Similar to the

single-variable delay embedding constructed based on Takens' embedding theory, the attractor manifold under generalized delay embedding also retains the dynamical information of the original system and also contains information about the future dynamics of time delays. However, unlike Takens' delay embedding, generalized delay embedding can have multiple different delay embedding manifolds at the same embedding dimension. These delay embedding manifolds reconstruct the dynamics of the same original high-dimensional system from different perspectives, so we collectively refer to them as multi-view delay embeddings. Note that another classic work based on the generalized delay embedding theorem is multi-view embedding (MVE). MVE generates multiple mappings that contain at least one non-delay variable as effective embeddings, and predicts the target variable through the neighborhood points on the attractor in the embedding space. Our method also considers multiple effective embeddings, but the prediction is completely completed through information transformation between delay manifolds and non-delay manifolds, without regard to the local topology of the attractor itself. This is the essential difference between our method and the classic method of MVE.

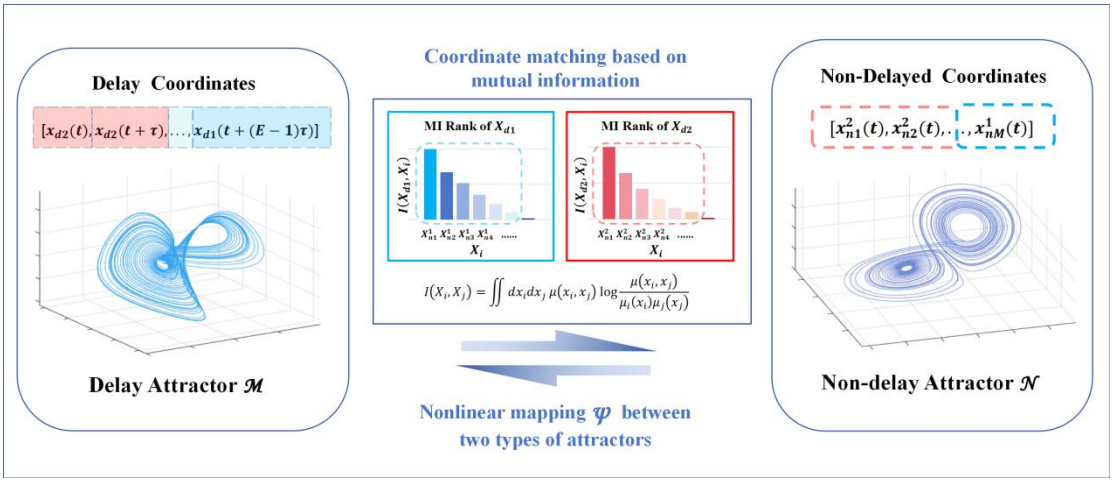

Fig.S1 Information transformation based on the isomorphism between manifolds.

1.3 Non-delayed attractors as predictors

In order to achieve the prediction of system dynamics, our goal is to directly predict the state points that contain future values in the multi-view delayed attractor. The multi-view delayed attractor and the original attractor manifold are diffeomorphic, and they maintain a topological isomorphism relationship between them. On the other hand, the non-delayed attractor based on the

generalized embedding theorem is also diffeomorphic to the original attractor manifold. The transfer of the isomorphism relationship implies that there is a smooth one-to-one mapping between the delayed attractor and the non-delayed attractor. If the one-to-one mapping between the two is determined, it is possible to predict the system dynamics after the observation time range by converting the state points on the attractor. For each view of the delayed attractor, our method uses strong correlation variables to construct a non-delayed embedding based on mutual information, and replaces the embedding manifold with the original attractor manifold as a predictor for the delayed attractor. The non-delayed part of each set of attractors undergoing information conversion retains the dynamical information of the original system in different forms, and has high correlation with the delayed part.

2. Reservoir Computing

Reservoir computing is a time-series oriented recurrent neural network variant model used in this work. This method evolved from Echo State Networks (ESNs)[4] and Liquid State Machines (LSMs)[5], alleviating the slow convergence rate and the vanishing and exploding gradients problems that exist in general RNN models. As a special type of recurrent neural network, reservoir computing uses a sparse randomly connected recurrent network to replace the intermediate hidden layer in the classical neural network architecture, thereby simplifying the complexity of the model. When using reservoir computing, we first construct a stochastic network with sparse topology, and complex functions can be achieved by training only a linear output layer during the model training phase. The reservoir computing model consists of three parts: an input layer, an output layer, and a reservoir for internal processing. Taking the *N*-dimensional observation data of an observation system as the input variable, a reservoir computing model with *M*-dimensional state variables, i.e. *M* internal neurons, and *L*-dimensional output variables has the following values at time *t* for the input unit, internal processing unit, and output unit:

$$\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_N(t)]^T$$

$$\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_M(t)]^T$$

$$\mathbf{y}(t) = [y_1(t), y_2(t), \ldots, y_L(t)]^T$$

The state equation and output equation of the RC model used in this article can be given by

the following equation:

$$\mathbf{x}(t+1) = \boldsymbol{f}\left(\boldsymbol{W}_{\text{res}}^{\text{res}}\,\mathbf{x}(t) + \boldsymbol{W}_{\text{inp}}^{\text{res}}\,\mathbf{u}(t) + \boldsymbol{W}_{\text{bias}}^{\text{res}}\right)$$
$$\hat{\mathbf{y}}(t+1) = \boldsymbol{W}_{\text{res}}^{\text{out}}\,\mathbf{x}(t+1) + \boldsymbol{W}_{\text{bias}}^{\text{out}}$$

Where, $\boldsymbol{f} = [f_1, f_2 ... f_M]$ represents the internal neuron activation function, and the hyperbolic tangent function is used here. All weight matrices to the reservoir $\boldsymbol{W}^{\text{res}}$ are randomly initialized with a uniform distribution and do not change after generation. The weight matrix to the output $\boldsymbol{W}^{\text{out}}$ needs to be trained using known observed time series data.

The training process of the RC network can be divided into two stages: the sampling stage and the weight computation stage. In the sampling stage, the initial state of the network is randomly selected, and the general zero initialisation is adopted here, i.e. $\mathbf{x}(0) = \mathbf{0}$. The input samples at time *t* are added to the reservoir through the input-reservoir weight matrix $\boldsymbol{W}_{\text{inp}}^{\text{res}}$, and the internal state of the reservoir and the model output are calculated and collected according to the state equation and output equation in turn. In this stage, the output variable **y**(t) and the internal state variable **x**(t) of the reservoir network are sampled, which are used to subsequently complete the calculation of the reservoir-output weight matrix. In the weight computation stage, the output weight matrix is calculated based on the internal state matrix of the reservoir and the expected target matrix obtained by connecting the variables collected during the sampling stage. Off-line training is used to calculate the output weight matrix, which is equivalent to solving the optimization problem $\min_{w} (\boldsymbol{X}w - \widetilde{\boldsymbol{Y}})^2$. The matrix $\boldsymbol{X}$ consists of all observed reservoir states, and the matrix $\widetilde{\boldsymbol{Y}}$ consists of all observed expected outputs at given time *t*. Tikhonov regression is used for regularization when optimizing the output weight matrix $\boldsymbol{W}^{\text{out}}$, and the optimal solution can be computed directly by the Moore-Penrose pseudo-inverse.

Based on the known information of the observation system and the final goal of the prediction task, the complete *N*-dimensional state variables of the observed system are used as input variables at the input end of the RC model, and the output variables at the output end of the model are vectorized prediction target matrices. With the MVIT framework, the output and input ends of the reservoir can be divided according to the coordinates of the selected delayed attractor and the corresponding non-delayed attractor, that is adopting parallel reservoir computing to efficiently predict high-dimensional systems. Parallel reservoir computing[13] uses a family of RC models in parallel, each model will be used to predict the evolution of a part of the system state, and all other

states that interact with this part of the state are provided at the input end of the reservoir network. This model utilizes local interactions between variables, using multiple small reservoirs instead of a large-scale reservoir network, thereby decoupling large-scale prediction tasks. After determining the input variable $\boldsymbol{u_i}(t)$ and the desired output matrix $\widetilde{\boldsymbol{Y}}_{\boldsymbol{i}}$ based on the attractor coordinates, the output weight matrix parameters of each RC model during model training are independently calculated according to an optimization problem with Tikhonov regularization.

3. Datasets Details

3.1 Coupled Lorenz System

To verify the ability of our method to capture the dynamics of high-dimensional nonlinear systems, we first consider a coupled Lorenz system as a benchmark. The i-th (i=1,2,...,N) coupled subsystem is given by the following equation:

$$\dot{x}_i = -\sigma[x_i - y_i + C\sum_{j=1}^{N} a_{ij}^{(x,y)}(y_j - y_i)]$$

$$\dot{y}_i = \rho x_i - y_i - x_i z_i,$$

$$\dot{z}_i = -\beta z_i + x_i y_i.$$

where the parameters are set to typical values, i.e., σ = 10, ρ = 28, β =8/3. The elements of the adjacency matrix a in the Lorenz system coupling network take on either a value of 0 or 1, depending on whether direct relationships have been established between subsystems. The adjacency matrix is defined as:

$$a_{ij} = \begin{cases} 1 & i < N, j = i + 1; \\ 1 & i = N, j = 1; \\ 0 & otherwise. \end{cases}$$

This implies that all variables within the system are coupled, and such coupling is transmitted through strong interactions between adjacent subsystems. The coupling strength is set at C = 0.01. In the experiment depicted in Figure 2 of the main text, N = 30, thus the system comprises a total of 90 variables. We generate random system initial values within the range [0, 10^{-6}], and sample data at time intervals of t = 0.01.

3.2 Lorenz-96 System

Another benchmark model considered in our work is the Lorenz-96 system[1], which serves a classic model to study nonlinear dynamics and chaotic phenomena. The model is introduced by Edward N. Lorenz in 1996 to simulate the interactions and evolution of atmospheric circulation.

The Lorenz96 model consists of a set of interacting variables, typically denoted as $x_1, x_2, ..., x_N$, represent different spatial positions or states within the system. The evolution of each variable is described by the following dynamical equation:

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F,$$

where i=1,...,N and $x_{-1} := x_{N-1}$ and $x_{N+1} := x_1$. $x_{i\pm1}$ represents the neighboring state of variable x_i, and F is an external driving force. This equation signifies that each variable is influenced by its neighboring variables and driven by an external force. It has been discovered that the Lorenz96 model exhibits unpredictable and highly chaotic behavior when the value of the driving force F is within a certain range[2,3].

In our work, we set the parameter F=8, which implies that the system's evolution will display complex chaotic phenomena. The total number of variables, denoted as N, can be adjusted to reflect the desired level of complexity in the model. Given the dimension N, we generate an N-dimensional time series with a time interval of Δt=0.01 using the fourth-order Runge-Kutta method based on the N-dimensional equations of the Lorenz96 system, and the multivariate time series served as our observation of the system.

4. Comparison Methods

We compare ALM with the following methods.

• **MA**: The Moving Average (MA) model is a commonly used time series analysis method in the fields of economics and statistics. It is utilized for forecasting and analyzing data that exhibit random fluctuations and trends over time.The MA model is based on the concept of moving averages, where past observed values are weighted and averaged to predict future values. It calculates the weighted average of these values to predict future values. The model assumes that the current error is a linear combination of past errors. The general form of the MA(q) model can be represented as:

$$y_t = \frac{y_{t-1} + y_{t-2} + \ldots + y_{t-q}}{q}.$$

where q denotes the number of past observed values used in the model.

• **ARIMA**[6]: The Autoregressive Integrated Moving Average (ARIMA) model is a widely used time series analysis method that combines autoregressive (AR), moving average (MA), and differencing components. It is a versatile model capable of capturing both the autoregressive and moving average properties of a time series, as well as handling non-stationary data through differencing. The general form of the ARIMA model is ARIMA(p, d, q):

$$\Delta d\, y_t = c + \phi_1 \Delta d\, y_{t-1} + \ldots + \phi_p \Delta d\, y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_q \varepsilon_{t-q},$$

where p represents the order of the autoregressive component, d represents the degree of differencing, and q represents the order of the moving average component. The autoregressive component (AR) models the linear relationship between the current value and a certain number of

past values, while the moving average component (MA) models the linear relationship between the current value and a certain number of past errors. The differencing component (I) is used to transform non-stationary data into stationary data by taking differences between consecutive observations.

• **VAR**[7]: The Vector Autoregressive (VAR) model is a multivariate time series analysis method used to analyze and forecast the interdependencies among multiple variables. It extends the Autoregressive (AR) model to multiple variables, allowing for a more comprehensive analysis of their relationships. The form of the VAR model can be represented as VAR(p):

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t,$$

where p represents the order of the lagged variables. In a VAR(p) model, each variable is regressed on its own lagged values and the lagged values of all other variables in the system. This allows for the modeling of dynamic relationships and feedback mechanisms between the variables.

• **VARM**[8]: The Vector Autoregressive Moving Average with Exogenous Variables (VARMAX) model is an extension of the Vector Autoregressive (VAR) model that incorporates both lagged variables and exogenous variables. The basic process of VARMAX includes the autoregressive process, the moving average process, and the independent exogenous terms (other unmodeled inputs). VARMAX is a multi-variable method and the basic form of VARMAX(p,q) is

$$y_t = c + \beta x_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \Theta_j \varepsilon_{t-j} + \varepsilon_t,$$

where c is a constant vector, ϕ_j is an autoregressive coefficients matrix, x_t is exogenous predictor variables, β is a regression coefficient matrix, Θ_j is a moving average matrix, and ε_t is random Gaussian innovation.

• **SVR**[9]: Support Vector Regression (SVR) is a powerful machine learning model used for regression tasks. It is an extension of Support Vector Machines (SVM) and is particularly effective when dealing with non-linear relationships between variables. SVR differs from traditional regression models by incorporating the concept of a margin. It aims to find a hyperplane that maximizes the margin while allowing for a certain level of error tolerance. The form of the SVR model can be represented as:

$$y = \omega^T x + b + \sum \alpha_i K(x_i, x)$$

Where y represents the predicted output, x is the input vector, $\omega^T x$ is the weight vector, b is

the bias term, $\alpha_i$is the Lagrange multiplier associated with each training sample x_i, and $K(x_i, x)$ is the kernel function, which measures the similarity between x_i and x.

• **LSTM**[10]: Long Short-Term Memory (LSTM) is currently one of the most widely used deep learning model in the field of time series analysis. The LSTM model consists of several key components, including the input gate, forget gate, output gate, and cell state. These components work together to control the flow of information and determine what information should be stored, forgotten, and outputted at each time step. LSTM model can be summarized as follows:

$$h_t, c_t = LSTM(x_t, h_{t-1}, c_{t-1})$$

In this equation, x_t represents the input at time step t, h_t is the hidden state at time step t, and c_t is the cell state at time step t. The hidden state and cell state jointly capture the information learned from the previous time steps and pass it to the current time step.

• **GRU**[11]: Gated Recurrent Unit (GRU) is a type of recurrent neural network that widely used in sequential data modeling tasks. It has fewer parameters compared to other RNN variants, such as LSTM, making it computationally more efficient. GRU model can be represented as follows:

$$z_t = \sigma(W_z[h_{t-1}, x_t])$$
$$r_t = \sigma(W_r[h_{t-1}, x_t])$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot g_t$$

where x_t represents the input at time step t, h_t is the hidden state at time step t, z_t is the update gate, r_t is the reset gate, and g_t is the candidate activation. W_z and W_r are the weight matrices associated with the update gate and reset gate, respectively. σ denotes the sigmoid activation function, and $\odot$ represents element-wise multiplication. The update gate determines how much of the previous hidden state should be combined with the candidate activation. It controls the flow of information from the previous time step to the current time step. The reset gate regulates how much of the previous hidden state should be forgotten. It decides which information from the past is relevant for the current time step.

• **tRC**: Traditional reservoir computing (tRC) was used as a standard reservoir computing model for comparison. The dynamic of the system is parameterized by a tRC model with a large sparse network as a reservoir, and multi-step prediction results are generated through the closed-loop evolution of tRC. The tRC model for comparison was set to the same hyperparameters with the MVIT-PRC model (except for the size of the reservoir network).

• **PRC**[13]: Parallel reservoir computing (PRC) uses a family of RC models in parallel, each model will be used to predict the evolution of a part of the system state. PRC was used as a comparison method in this work to evaluate the predictive potential of the MVIT framework. The parallelized reservoir computing models generate local predictions of subsystem states and generate multi-step prediction results through closed-loop evolution. The PRC model for comparison was set to the same hyperparameters with the MVIT-PRC model.

• **ALM**[12]: Anticipated Learning Machine (ALM) is a novel deep learning method that combines delayed embedding theory and neural network framework.The core of the ALM model is the spatial-temporal information-transformation equation (STI equation), which is based on Takens'

embedding theorem. Through the STI equation, ALM transforms the recent information into the future dynamical system of the target variables. ALM can simulate a large number of randomly selected non-delayed attractors using the Dropout strategy to train AL neural networks to accurately and robustly reconstruct the system dynamics and make multi-step predictions through iterative training. Based on the nonlinear dynamics rather than the traditional statistics, ALM represents a new paradigm of machine learning which is dynamics-oriented.

5. Supplemental Figures

5.1 The prediction of the coupled Lorenz system

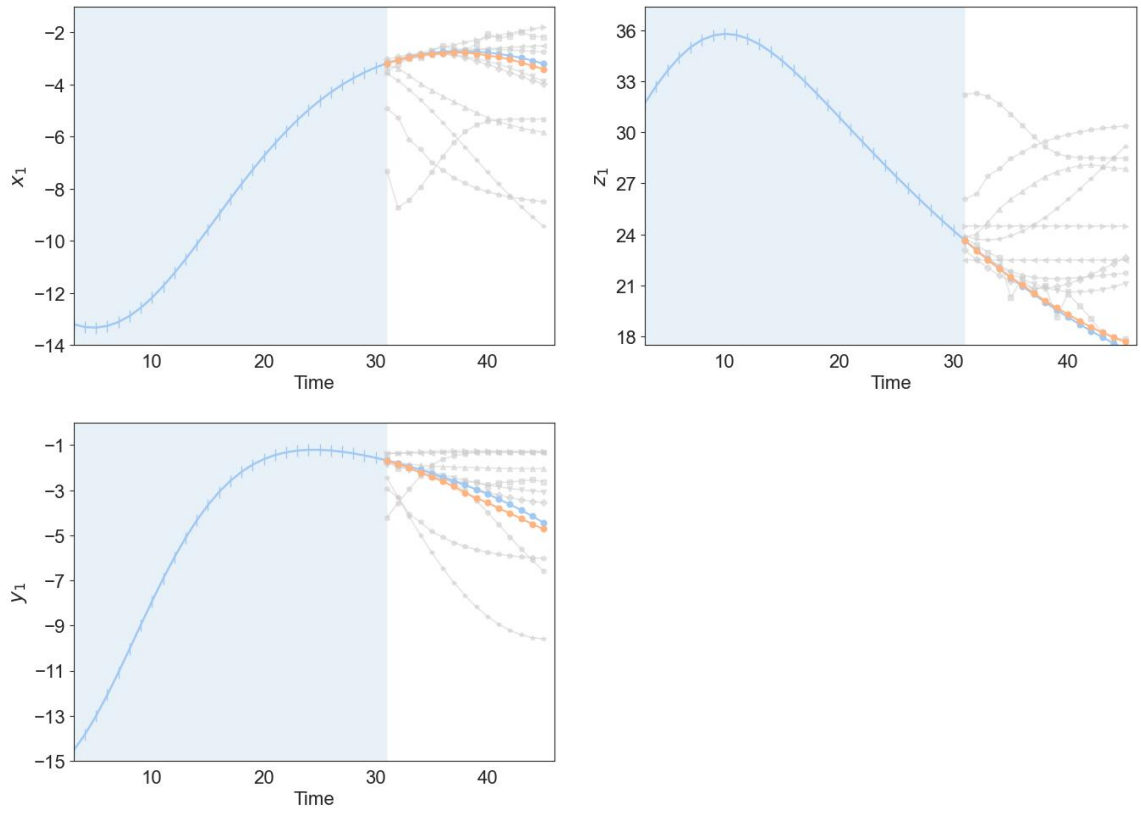

Fig.S2 Our framework achieves 15 step prediction results on a 90 dimensional coupled Lorenz system. We train a prediction model with a 15th-order delay using 30 observations of the system as known information. There are only 15 valid sample pairs available for learning. Our multi perspective information conversion mechanism effectively expands available information in this data shortage situation.

5.2 The prediction of San Francisco Traffic Dataset

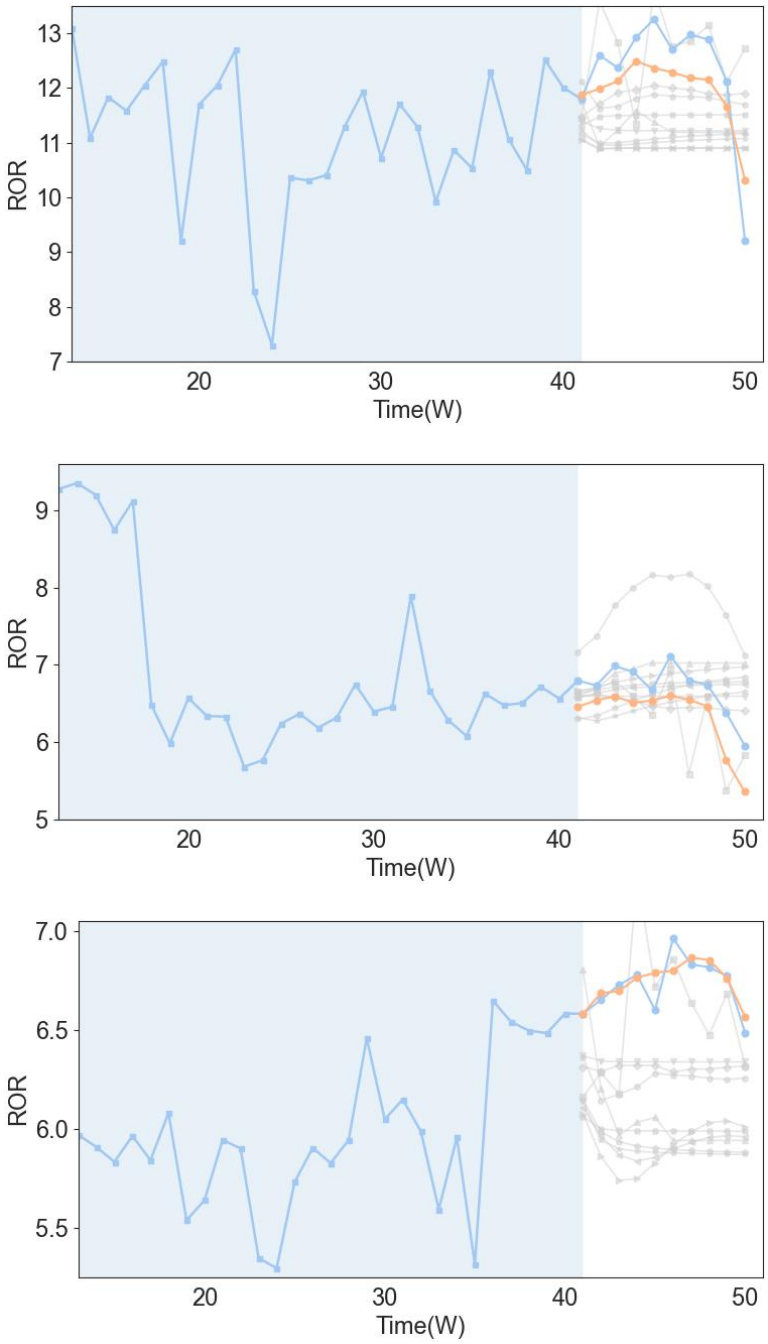

Fig.S3 Synchronous prediction results of multi-view joint prediction model for the other three dimensions on San Francisco traffic dataset.

5.3 The prediction of NN5 Dataset

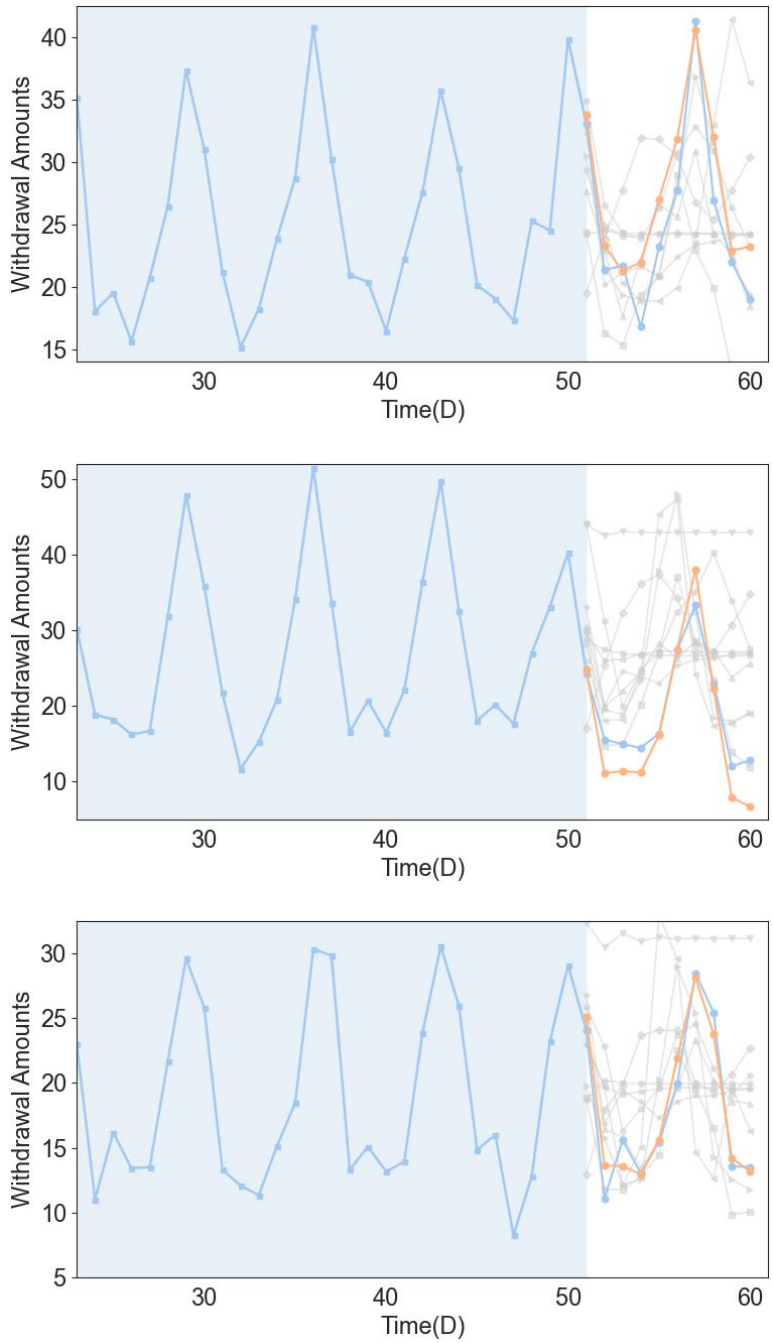

Fig.S4 Synchronous prediction results of multi-view joint prediction model for the other three dimensions on NN5 dataset.

5.4 The prediction of Electricity Dataset

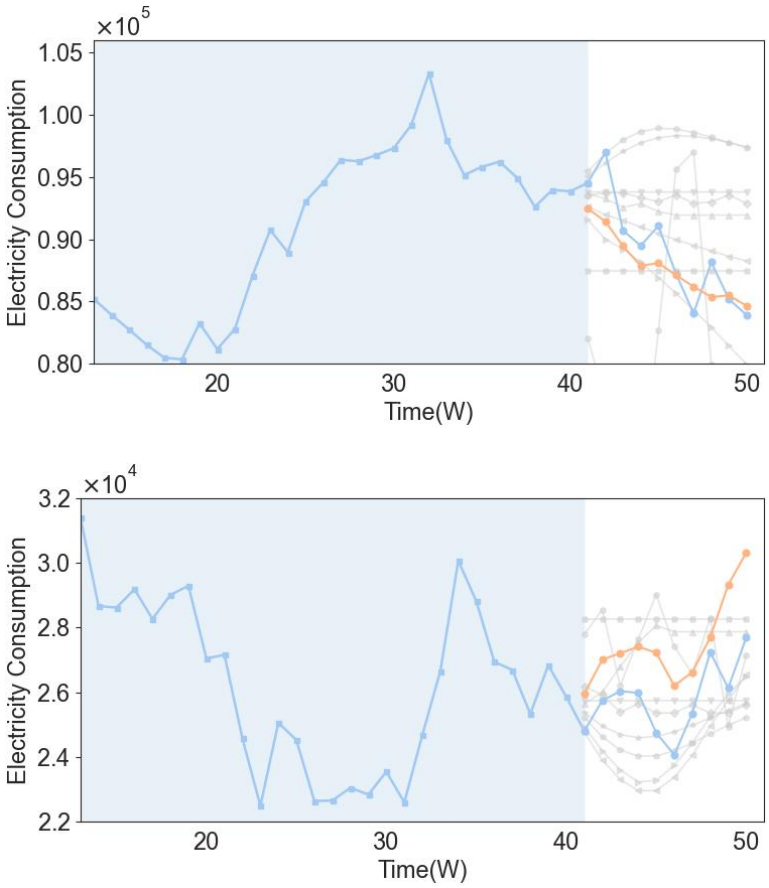

Fig.S5 Synchronous prediction results of multi-view joint prediction model for the other two dimensions on Electricity dataset.

5.5 The prediction of Solar Energy Dataset

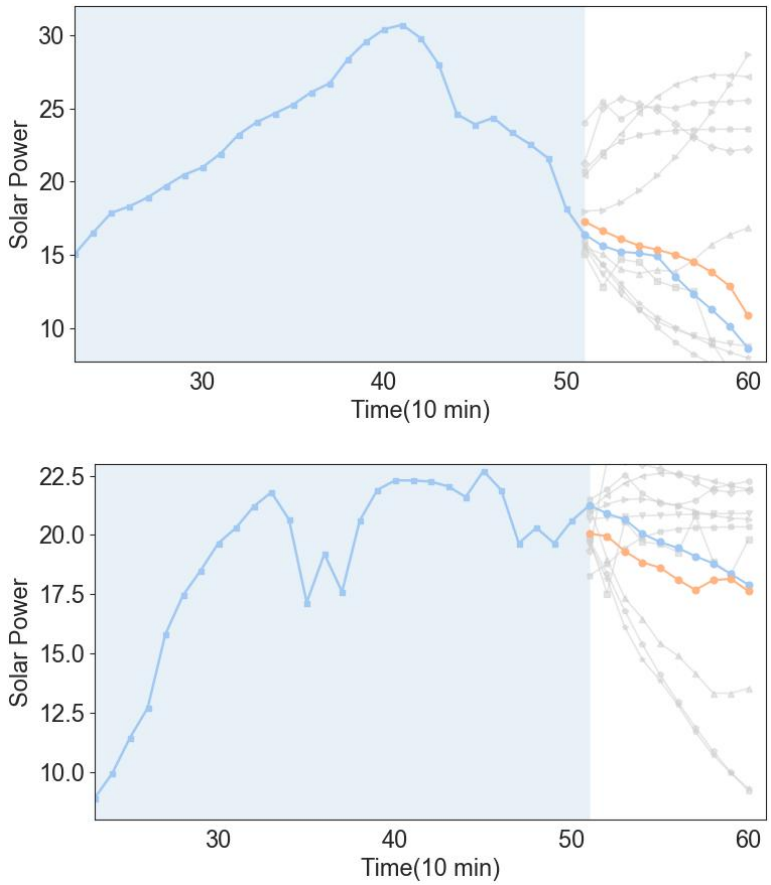

Fig.S6 Synchronous prediction results of multi-view joint prediction model for the other two dimensions on Solar Energy dataset.

5.6 The prediction of Hospital Dataset

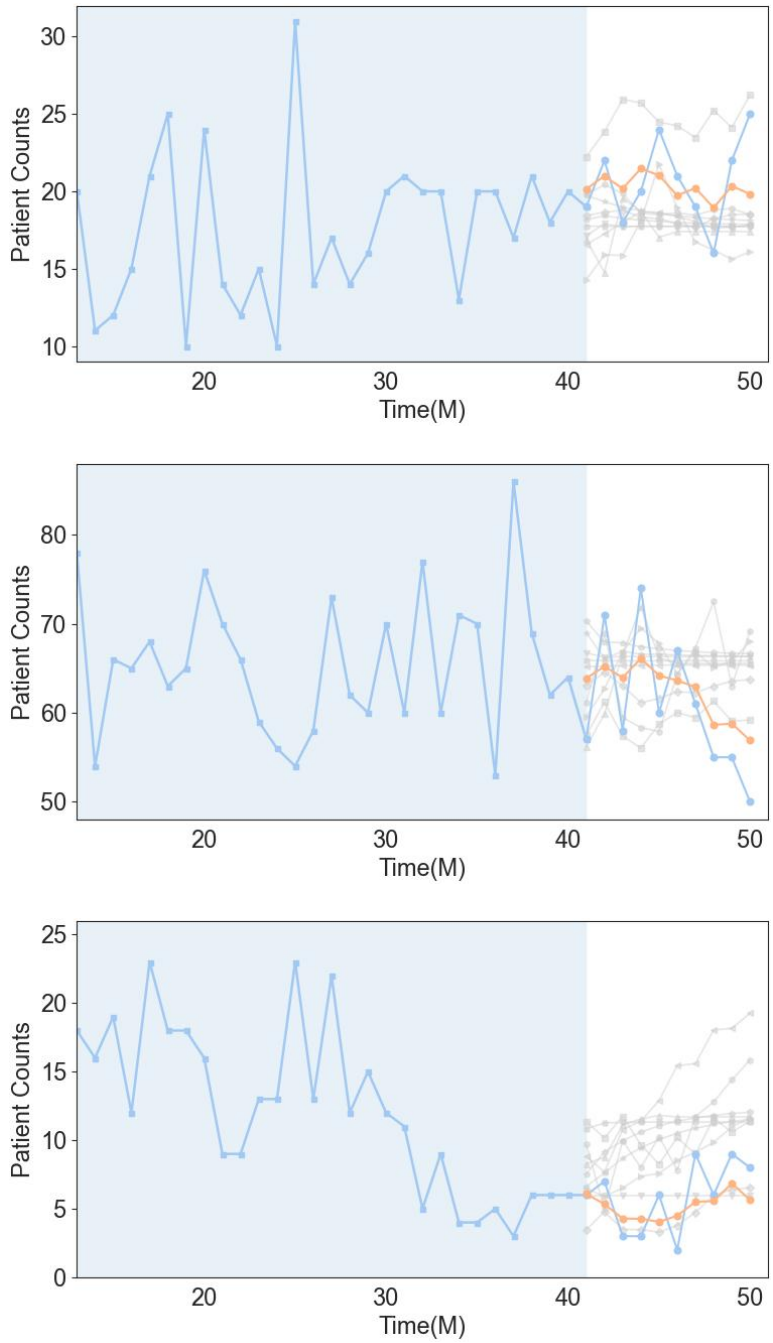

Fig.S7 Synchronous prediction results of multi-view joint prediction model for the other three dimensions on Hospital dataset.

5.7 Comparison between Multi-View and Single-View

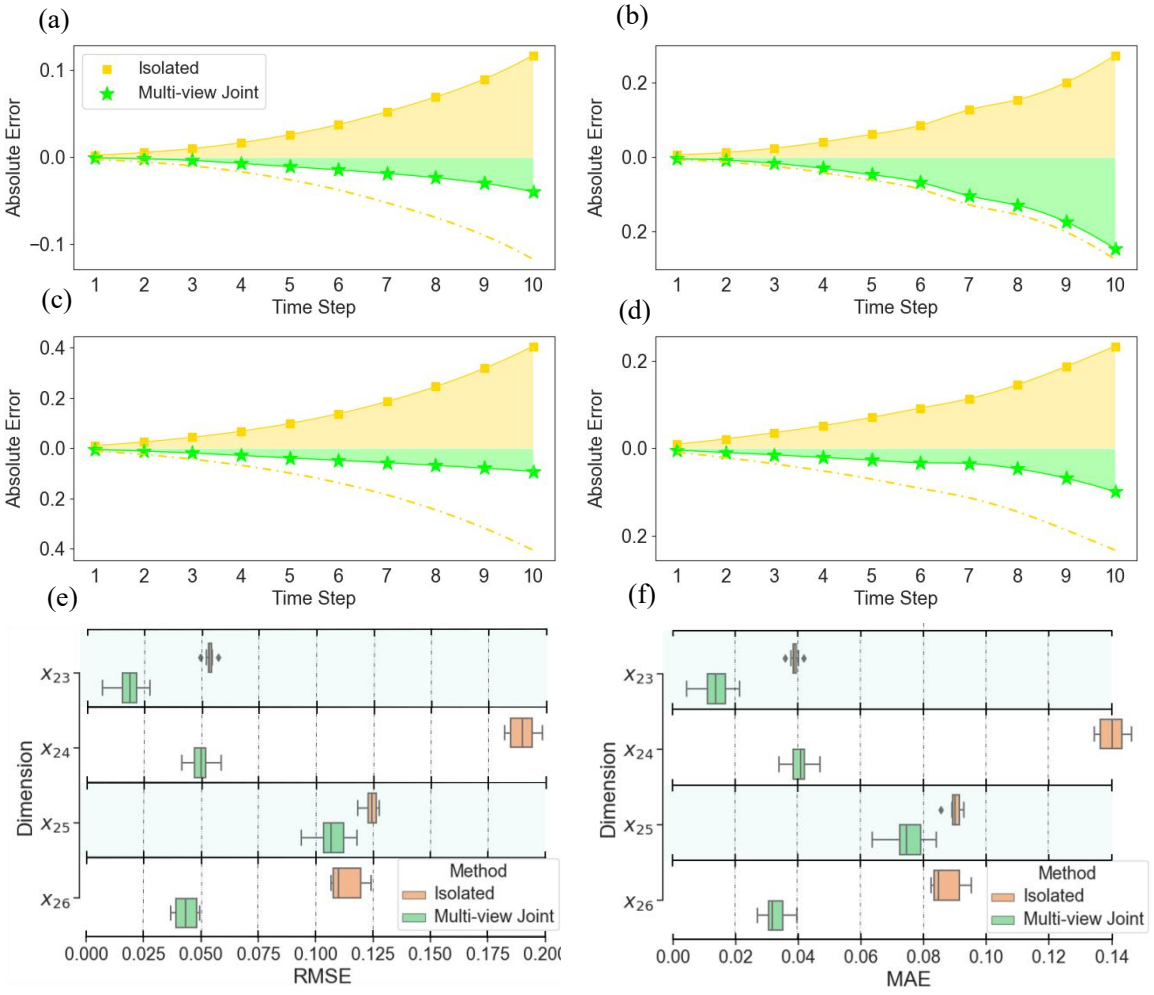

Fig.S8 Comparison of prediction performance between multi-view models and separate single-view models in the Lorenz 96 system. (a-d) The absolute errors between the mean of the final predictions and the true observations of two kinds of models on three dimensions of a subsystem in a 144-dimensional Lorenz 96 system. (e-f) the box plot of RMSEs and MAEs for 50 independent experiments.

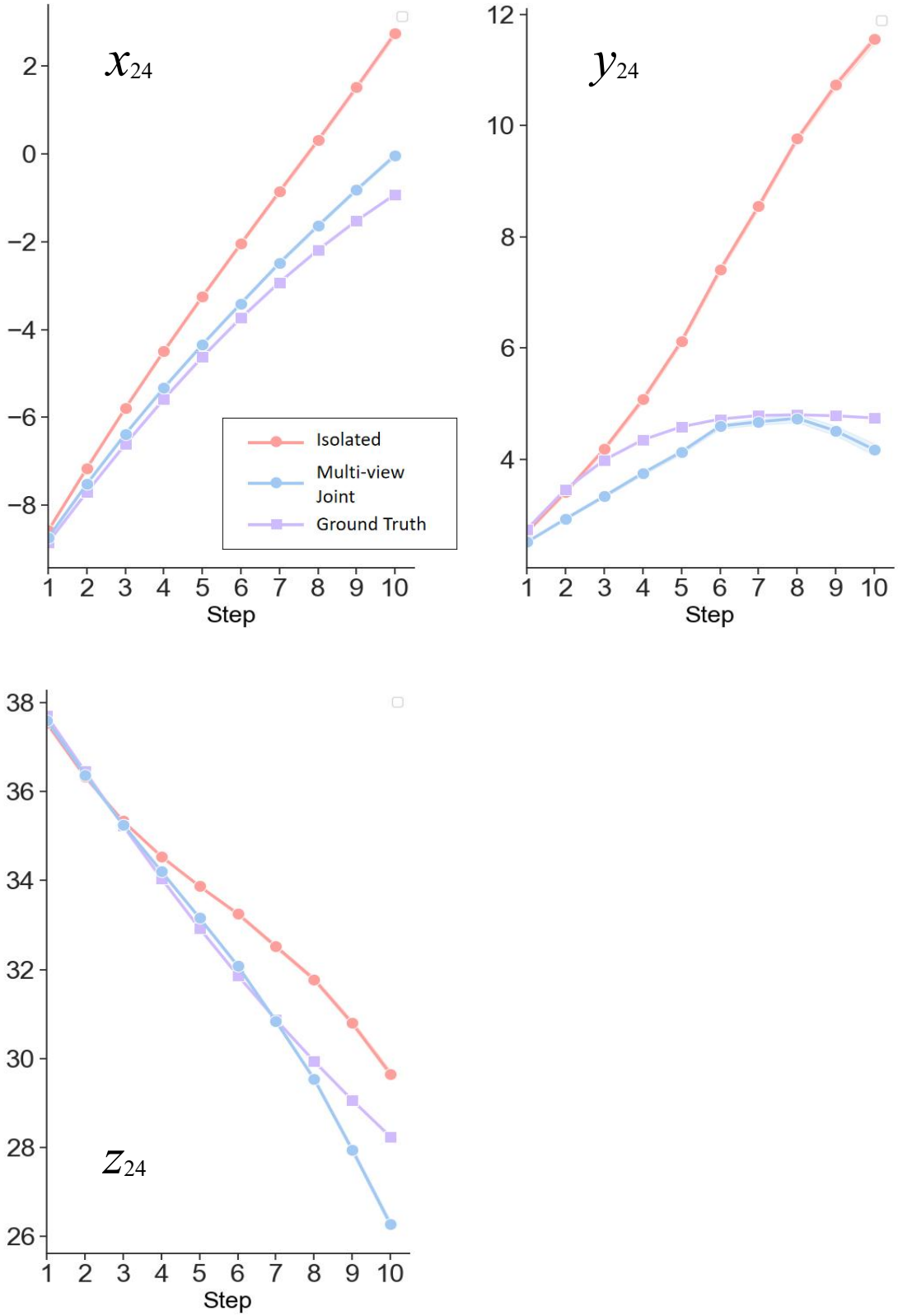

Fig.S8 Comparison of prediction performance between multi-view models and separate single-view models. The prediction results of two kinds of models on three dimensions of a subsystem in a 90-dimensional coupled Lorenz system.

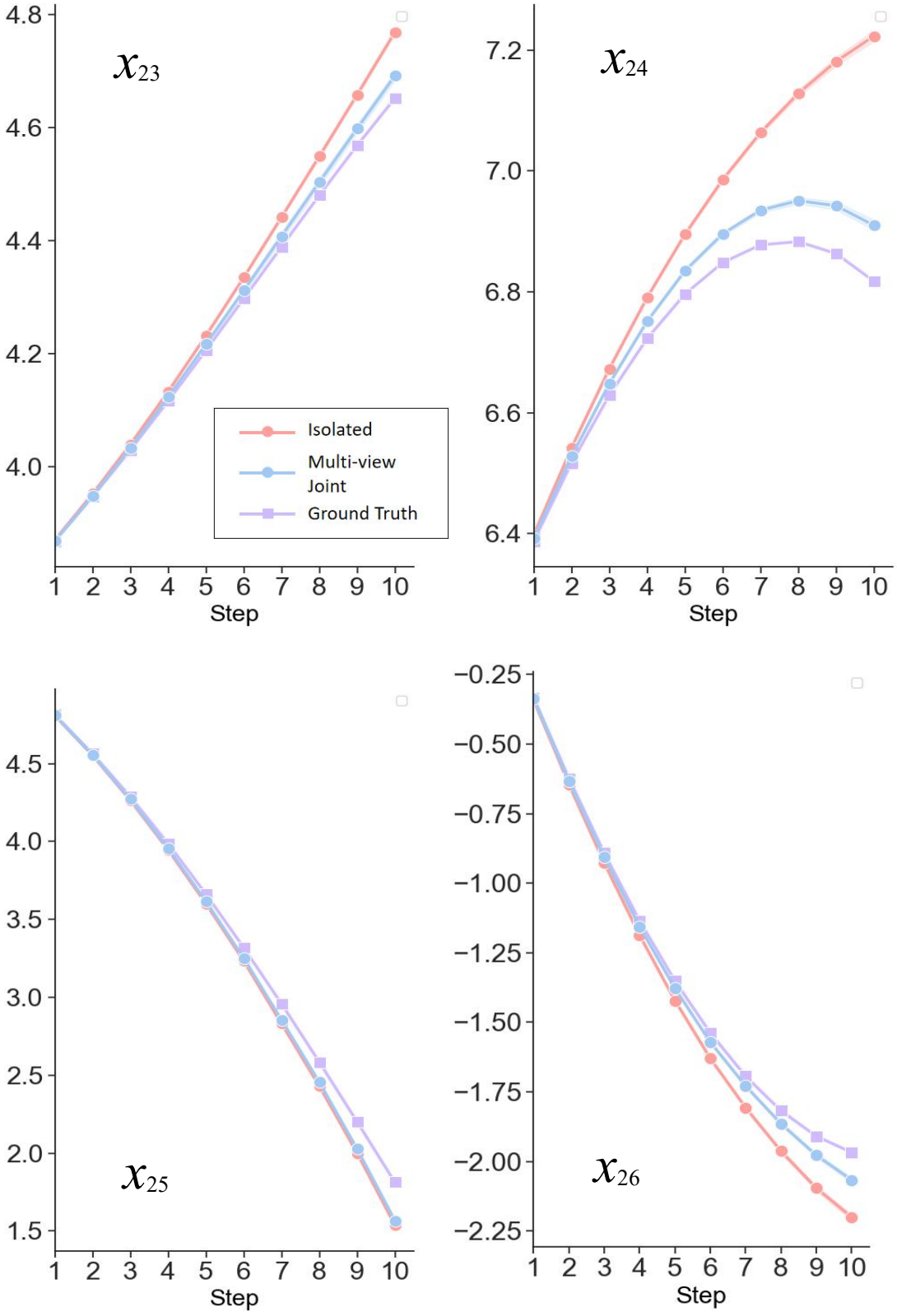

Fig.S9 Comparison of prediction performance between multi-view models and separate single-view models. The prediction results of two kinds of models on four dimensions of a subsystem in a 144-dimensional Lorenz-96 system.

6. Supplemental Tables

6.1 The estimated box dimension in datasets

Table S1 The estimated box dimension in selected systems

Dataset	Number of observed variables	Box dimension (d)
The 90-dimensional coupled Lorenz systems	90	2.62
The 144-dimensional Lorenz-96 systems	144	3.27
San Francisco Traffic	862	4.78
NN5 Dataset	111	3.67
Electricity Dataset	321	1.34
Solar Energy Dataset	137	2.06
Hospital Dataset	767	3.09

The box-counting dimensions are approximately estimated by using the R package “Rdimtools”

6.2 The performances of prediction methods on the 90D coupled Lorenz system

Table S2 Results on 3 dimensions within a subsystem (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.013	0.044	0.734	1.162	0.786	0.08	0.176	0.092	0.21	0.37	1.177
0.018	0.037	0.133	0.476	0.767	0.037	0.017	0.096	0.267	0.189	0.288
0.034	0.197	1.555	2.107	1.067	0.203	0.497	0.405	2.417	1.138	2.604
Average										
0.022	0.092	0.807	1.249	0.873	0.107	0.23	0.198	0.964	0.566	1.356

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.01	0.032	0.627	1.113	0.688	0.059	0.136	0.056	0.15	0.331	1.141
0.015	0.025	0.104	0.466	0.703	0.03	0.016	0.06	0.248	0.162	0.255
0.031	0.14	1.301	1.967	0.878	0.17	0.367	0.231	2.383	1.029	2.602
Average										
0.019	0.066	0.677	1.182	0.756	0.086	0.173	0.116	0.927	0.507	1.333

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.991	0.683	-0.883	-0.947	-0.82	0.861	-0.55	0.247	0.791	-0.81	0.656
0.999	0.994	0.876	0.946	0.984	0.945	0.998	0.851	-0.588	-0.864	-0.9
0.999	0.985	-0.992	-0.984	-0.886	0.893	0.473	0.215	-0.623	-0.008	0.977
Average										
0.997	0.887	-0.333	-0.328	-0.241	0.9	0.307	0.437	-0.14	-0.561	0.244

*The performance metrics include the values of the root mean square error (RMSE), mean absolute error (MAE) and the Pearson correlation coefficient (PCC). The RMSE and MAE was normalized by the standard deviation of the known observed data (the same applied to **Table S3**-**Table S12**).

Table S3 Results on 3 dimensions within a subsystem (E=15+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.029	0.095	0.566	1.283	1.115	0.155	0.122	0.051	0.077	0.183	1.032
0.056	0.116	0.241	0.457	0.889	0.159	0.066	0.206	0.355	0.364	0.383
0.05	0.497	2.053	2.62	1.836	0.198	0.691	0.641	0.85	1.325	2.72
Average										
0.045	0.236	0.953	1.453	1.28	0.171	0.293	0.299	0.427	0.624	1.379

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.022	0.072	0.511	1.241	0.984	0.119	0.095	0.035	0.052	0.134	0.977
0.047	0.079	0.186	0.447	0.833	0.106	**0.045**	0.136	0.31	0.319	0.343
0.035	0.358	1.82	2.451	1.509	0.169	0.497	0.481	0.711	1.205	2.714
Average										
0.035	0.169	0.839	1.38	1.109	0.132	0.212	0.217	0.358	0.553	1.345

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.942	0.721	-0.062	-0.264	0.102	0.185	0.552	0.624	0.139	0.001	0.112
0.997	0.967	0.842	0.876	0.935	0.729	0.98	0.992	-0.732	-0.652	-0.763
0.999	0.877	-0.947	-0.964	-0.933	0.954	0.167	0.776	-0.544	0.856	0.957
Average										
0.98	0.855	-0.056	-0.118	0.035	0.623	0.566	0.797	-0.379	0.068	0.102

Table S4 Results on 3 dimensions within a subsystem (E=20+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.037	0.136	0.326	1.303	1.329	0.268	0.841	0.45	0.258	0.392	0.923
0.069	0.263	0.494	0.404	0.867	0.309	0.585	0.091	0.386	0.466	0.535
0.133	0.89	2.147	2.997	2.568	0.275	1.492	0.297	1.459	1.582	2.893
Average										
0.079	0.429	0.989	1.568	1.588	0.284	0.973	0.279	0.701	0.813	1.45

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.029	0.108	0.297	1.27	1.194	0.202	0.586	0.312	0.216	0.312	0.847
0.047	0.177	0.403	0.369	0.821	0.216	0.455	0.062	0.288	0.379	0.465
0.114	0.653	1.911	2.816	2.137	0.23	0.99	0.196	1.303	1.439	2.874
Average										
0.063	0.313	0.87	1.485	1.384	0.216	0.68	0.19	0.602	0.71	1.396

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.988	0.949	0.367	0.333	0.666	-0.421	0.921	0.932	-0.024	-0.615	-0.326
0.987	0.893	-0.851	0.792	0.811	0.725	0.996	0.956	0.472	0.308	-0.647
0.999	0.396	-0.912	-0.945	-0.949	0.956	-0.56	0.95	-0.422	0.869	0.919
Average										
0.991	0.746	-0.465	0.06	0.176	0.42	0.452	0.946	0.009	0.187	-0.018

6.3 The performances of prediction methods on the 90D coupled Lorenz system with noise

Table S5 Results on 3 dimensions within a subsystem (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.165	0.469	0.936	0.681	0.42	**0.13**	0.41	0.168	0.29	0.559	1.119
0.062	0.146	0.178	0.6	0.459	0.191	0.11	0.193	0.341	0.251	0.277
0.093	0.773	1.871	0.742	0.925	0.466	1.436	0.921	1.779	0.594	2.555
Average										
0.107	0.463	0.995	0.674	0.601	0.262	0.652	0.427	0.804	0.468	1.317

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.132	0.462	0.914	0.632	0.369	**0.087**	0.366	0.125	0.288	0.524	1.118
0.048	0.117	0.115	0.56	0.431	0.173	0.097	0.158	0.326	0.234	0.245
0.065	0.701	1.712	0.635	0.834	0.438	1.296	0.823	1.615	0.549	2.533
Average										
0.082	0.427	0.914	0.609	0.545	0.233	0.586	0.369	0.743	0.436	1.299

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.742	0.179	0.057	-0.843	-0.703	0.669	-0.347	0.061	-0.535	0.738	-0.133
0.975	-0.414	0.228	0.989	0.98	0.535	0.78	0.765	-0.873	0.972	-0.354
0.99	0.322	-0.974	-0.485	-0.535	0.811	-0.939	-0.522	-0.945	0.549	0.235
Average										
0.902	0.029	-0.23	-0.113	-0.086	0.672	-0.169	0.102	-0.784	0.753	-0.084

6.4 The performances of prediction methods on the 144D Lorenz-96 system

Table S6 Results on 4 dimensions within a subsystem (E=15+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.079	0.637	1.547	1.229	1.21	0.389	0.241	0.184	1.644	1.746	1.582
0.034	0.236	0.439	1.031	1.107	0.129	0.068	0.052	0.346	0.283	0.079
0.027	0.936	2.363	3.101	3.176	0.111	0.162	0.039	0.355	0.544	1.9
0.068	1.503	3.474	4.164	4.209	0.246	0.245	0.759	1.206	1.726	3.356
Average										
0.052	0.765	1.778	2.407	2.478	0.21	0.166	0.196	0.765	0.886	1.416

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.059	0.504	1.32	1.087	1.069	0.314	0.22	0.122	1.487	1.424	1.421
0.026	0.19	0.409	1.019	1.084	0.088	0.051	0.033	0.312	0.248	0.072
0.02	0.706	2.126	3.023	3.1	0.095	0.158	0.028	0.283	0.513	1.821
0.058	1.17	2.941	3.807	3.85	0.2	0.223	0.496	1.037	1.224	2.979
Average										
0.04	0.594	1.566	2.296	2.361	0.162	0.149	0.129	0.671	0.723	1.301

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.999	0.982	-0.934	0.87	0.629	0.968	0.999	0.998	-0.898	-0.984	-0.673
0.998	-0.961	0.796	0.697	0.455	0.986	**0.999**	0.996	-0.893	-0.886	0.605
0.999	0.968	-0.971	-0.9	-0.719	0.992	0.998	0.999	0.995	0.999	-0.804
0.999	0.986	-0.916	-0.794	-0.583	0.996	0.997	0.981	0.752	0.522	-0.631
Average										
0.999	0.494	-0.506	-0.032	-0.055	0.985	0.998	0.994	-0.011	-0.087	-0.376

6.5 The performances of prediction methods on the 144D Lorenz-96 system with noise

Table S7 Results on 4 dimensions within a subsystem (E=15+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.734	1.203	3.103	1.889	1.044	0.779	0.804	1.67	2.344	1.921	0.895
0.318	0.605	0.56	2.184	0.886	0.46	0.694	1.362	0.587	0.738	0.767
0.386	2.326	2.455	4.06	3.234	1.163	2.079	2.858	2.299	3.304	3.45
1.187	3.091	3.69	2.827	2.963	1.374	1.47	2.498	5.144	3.908	3.062
Average										
0.612	1.747	2.156	3.249	2.136	0.964	1.395	2.323	2.245	2.364	2.124

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.611	1.091	2.52	1.51	0.896	0.699	0.673	1.511	1.822	1.85	0.736
0.209	0.6	0.373	2.133	0.87	0.452	0.675	1.299	0.519	0.641	0.755
0.368	2.259	2.299	3.964	3.162	1.109	2.003	2.749	2.259	2.84	3.422
0.822	2.719	3.369	2.32	2.535	1.266	1.278	2.207	4.651	3.645	2.65
Average										
0.463	1.645	1.856	3.048	2.011	0.914	1.309	2.18	2.03	2.114	2.016

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.762	-0.094	0.332	-0.758	-0.78	0.684	0.563	0.432	0.145	0.646	-0.478
0.434	0.341	0.175	-0.535	0.441	0.457	**0.781**	0.427	0.011	-0.094	-0.451
0.936	0.2	0.068	-0.156	-0.701	0.658	-0.407	-0.142	0.766	-0.142	0.69
0.792	0.297	0.106	-0.163	-0.651	0.445	0.784	0.656	0.458	0.433	-0.326
Average										
0.731	0.186	0.17	-0.403	-0.423	0.561	0.430	0.343	0.345	0.211	-0.141

6.6 The performances of prediction methods on San Francisco Traffic Dataset

Table S8 Results in 4 locations (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.236	1.22	1.385	1.247	1.215	1.08	0.565	1.212	1.387	1.392	1.062
0.355	0.307	0.399	0.413	0.426	0.491	0.331	1.016	0.308	0.367	**0.307**
0.545	1.345	1.325	1.452	1.487	1.136	1.017	1.049	1.554	1.559	1.2
0.192	0.949	1.71	1.906	1.953	0.774	1.026	1.171	1.897	1.93	1.737
Average										
0.332	0.955	1.205	1.254	1.27	0.87	0.735	1.112	1.286	1.312	1.076

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.187	1.118	1.292	1.147	1.115	0.882	0.37	1.073	1.287	1.266	0.986
0.312	0.262	0.281	0.343	0.389	0.34	0.289	0.968	0.23	0.257	**0.221**
0.461	1.275	1.232	1.396	1.429	0.699	0.836	0.916	1.504	1.509	1.112
0.127	0.892	1.621	1.859	1.907	0.671	0.971	1.115	1.852	1.879	1.703
Average										
0.272	0.887	1.106	1.186	1.21	0.648	0.616	1.018	1.218	1.228	1.006

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.921	-0.247	0.125	-0.324	-0.209	0.273	0.331	-0.608	0.037	0.093	0.431
0.889	0.094	-0.209	-0.496	-0.553	0.678	0.445	0.567	-0.32	-0.481	-0.335
0.965	-0.122	0.032	-0.457	-0.377	0.18	0.235	0.176	-0.173	-0.162	0.109
0.8	-0.369	-0.427	-0.377	-0.376	0.519	-0.544	-0.115	-0.348	-0.099	-0.368
Average										
0.894	-0.161	-0.12	-0.413	-0.379	0.412	0.117	0.005	-0.201	-0.162	-0.041

6.7 The performances of prediction methods on Solar Energy Dataset

Table S9 Results on 3 sites (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.033	0.105	0.216	0.726	0.651	0.186	0.684	0.636	0.727	0.797	0.253
0.125	0.206	0.506	0.767	0.79	0.176	0.373	0.311	0.353	0.216	0.213
0.172	0.289	0.391	0.352	0.27	0.213	1.108	1.278	1.34	1.112	1.074
Average										
0.20	0.343	0.371	0.615	0.57	0.192	0.722	0.742	0.807	0.708	0.514

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.024	0.086	0.21	0.638	0.572	0.165	0.676	0.557	0.701	0.776	0.216
0.11	0.169	0.467	0.708	0.738	0.114	0.367	0.273	0.318	0.194	0.185
0.144	0.252	0.28	0.321	0.243	0.177	1.085	1.242	1.248	0.91	1.021
Average										
0.092	0.169	0.319	0.556	0.518	0.152	0.709	0.691	0.755	0.627	0.474

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.953	-0.373	0.818	0.919	0.905	0.113	0.479	-0.494	-0.811	-0.414	-0.866
0.886	-0.959	0.937	**0.992**	0.986	0.003	-0.081	-0.17	-0.418	0.645	-0.877
0.982	0.833	-0.701	0.9	0.897	0.951	0.52	-0.677	-0.811	-0.993	-0.683
Average										
0.94	-0.166	0.351	0.937	0.929	0.356	0.306	-0.447	-0.68	-0.254	-0.809

6.8 The performances of prediction methods on NN5 Dataset

Table S10 Results of 4 daily withdrawal amounts (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.385	2.851	0.97	1.177	1.045	0.641	1.755	1.605	1.447	1.223	1.177
0.164	1.798	0.457	0.721	0.678	0.609	1.001	0.7	1.184	0.68	0.778
0.51	3.301	0.747	1.064	1.059	1.322	1.56	0.778	1.73	**0.407**	1.146
0.112	0.325	0.245	0.313	0.266	0.229	0.492	0.257	0.23	0.477	0.347
Average										
0.293	2.069	0.605	0.819	0.762	0.7	1.202	0.835	1.148	0.697	0.862

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.292	2.743	0.866	1.049	0.935	0.474	1.557	1.42	1.082	0.931	1.052
0.122	1.657	0.345	0.636	0.611	0.484	0.958	0.601	0.979	0.547	0.717
0.401	3.117	0.592	0.792	0.704	1.088	1.336	0.711	1.307	**0.308**	0.877
0.075	0.281	0.209	0.271	0.214	0.196	0.453	0.21	0.196	0.403	0.303
Average										
0.223	1.95	0.503	0.687	0.616	0.56	1.076	0.736	0.891	0.547	0.737

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.986	0.261	0.6	0.372	0.318	0.803	-0.373	0.555	0.323	0.591	0.135
0.979	0.457	0.811	0.376	0.421	0.776	-0.368	0.605	-0.112	0.562	-0.433
0.966	0.4	0.821	0.377	0.445	0.609	-0.48	0.884	-0.015	0.938	-0.268
0.972	0.429	0.698	0.583	0.597	0.756	-0.549	0.708	0.722	-0.31	-0.622
Average										
0.976	0.387	0.732	0.427	0.445	0.736	-0.442	0.688	0.23	0.445	-0.297

6.9 The performances of prediction methods on Electricity Dataset

Table S11 Results on 4 customers (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.375	0.879	0.847	0.629	0.623	0.966	0.724	1.359	0.814	0.653	1.747
0.369	0.976	0.767	1.555	1.506	1.392	0.902	3.108	0.498	0.583	0.696
0.508	**0.296**	0.555	0.432	0.333	0.706	0.32	0.649	0.502	0.439	0.759
0.698	1.343	1.886	1.892	1.894	0.699	2.021	2.523	1.299	2.088	2.004
Average										
0.488	0.874	1.014	1.127	1.089	0.941	0.992	1.91	0.778	0.941	1.301

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.275	0.679	0.754	0.52	0.516	0.964	0.517	1.066	0.632	0.5	1.651
0.278	0.854	0.661	1.371	1.322	1.390	0.79	2.755	0.419	0.497	0.557
0.438	**0.235**	0.438	0.371	0.281	0.705	0.265	0.541	0.428	0.364	0.699
0.558	1.207	1.716	1.717	1.696	0.693	1.814	1.977	1.146	1.865	1.826
Average										
0.388	0.744	0.892	0.995	0.954	0.938	0.846	1.585	0.656	0.807	1.183

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.96	-0.498	0.789	-0.371	-0.473	-0.296	0.505	-0.191	0.94	0.936	0.355
0.892	-0.404	0.862	-0.498	-0.72	0.541	0.581	0.192	0.892	0.871	0.436
0.722	0.304	0.193	0.436	0.524	-0.554	-0.136	-0.17	0.643	0.627	0.308
0.781	-0.745	-0.873	-0.861	-0.832	0.314	0.15	0.172	-0.055	-0.914	-0.351
Average										
0.839	-0.336	0.243	-0.323	-0.375	0.001	0.275	0.001	0.605	0.38	0.187

6.10 The performances of prediction methods on Hospital Dataset

Table S12 Results on 4 research targets (E=10+1)

(a) RMSE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.443	0.645	1.03	0.892	0.763	0.836	0.608	0.742	1.739	1.502	1.075
0.33	0.421	1.022	0.954	0.811	0.929	0.392	0.946	1.544	0.585	1.036
0.619	1.11	1.035	1.192	1.153	0.95	0.956	1.249	1.036	1.098	1.071
0.517	0.864	1.011	0.816	0.752	1.073	0.767	0.721	0.832	1.002	0.864
Average										
0.477	0.76	1.024	0.963	0.87	0.947	0.681	0.914	1.288	1.047	1.011

(b) MAE

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.309	0.547	0.885	0.759	0.653	0.631	0.477	0.66	1.52	1.344	0.941
0.275	0.338	0.9	0.848	0.685	0.794	0.328	0.891	1.394	0.479	0.946
0.555	0.992	0.867	1.042	1.03	0.706	0.828	0.96	0.938	0.962	0.964
0.426	0.725	0.845	0.661	0.608	0.881	0.64	0.59	0.681	0.823	0.715
Average										
0.391	0.651	0.874	0.828	0.744	0.753	0.568	0.775	1.133	0.902	0.892

(c) PCC

MVPRC	ARIMA	MA	VAR	VARM	ALM	tRC	PRC	LSTM	GRU	SVR
0.932	0.262	0.46	0.551	0.604	0.433	0.62	0.812	0.312	0.167	0.223
0.718	0.013	-0.048	0.226	0.282	0.189	0.675	0.548	0.356	0.495	-0.026
0.821	-0.253	0.204	0.098	-0.194	0.066	-0.337	-0.265	-0.14	0.04	-0.136
0.414	0.257	-0.342	-0.116	-0.195	0.31	0.14	-0.213	0.225	0.384	-0.106
Average										
0.721	0.07	0.068	0.19	0.124	0.249	0.275	0.221	0.188	0.271	-0.011

References

[1] Lorenz, E. N. (1996, September). Predictability: A problem partly solved. In *Proc. Seminar on predictability* (Vol. 1, No. 1).

[2] Lorenz, E. N., & Emanuel, K. A. (1998). Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, *55*(3), 399-414.

[3] Brajard, J., Carrassi, A., Bocquet, M., & Bertino, L. (2020). Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model. *Journal of computational science*, *44*, 101171.

[4] Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: Predicting chaotic systems andsaving energy in wireless communication. *Science* 304, 78−80.

[5] Maass, W., Natschläger, T., Markram, H., 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput*. 14, 2531−2560.

[6] Box, G. E. & Pierce, D. A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association* 65, 1509-1526 (1970).

[7] Lütkepohl, H. Vector autoregressive and vector error correction models. Applied time series econometrics (2004).

[8] Lütkepohl, H. New introduction to multiple time series analysis. (Springer Science & Business Media, 2005).

[9] Vapnik, V. The nature of statistical learning theory. (Springer science & business media, 2013).

[10] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* 9, 1735-1780 (1997).

[11] Cho K , Van Merrienboer B , Gulcehre C ,et al.Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014. DOI: 10.3115 /v1/D14-1179.

[12] C. Chen et al., Predicting future dynamics from short-term time series by anticipated learning machine, National Science Review, Journal vol. 6, 2020.

[13] Pathak, J., Hunt, B., Girvan, M., Lu, Z., Ott, E., 2018. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. Phys.Rev. Lett. 120, 024102.