SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2026, Vol. 69, Iss. 7, 172201:1–172201:15
https://doi.org/10.1007/s11432-024-4836-1

Spatiotemporal diffusion with Koopman operator for multistep prediction of short-term time-series

Liangyu SU[1], Jun SHU[1,2*], Luonan CHEN[3], Deyu MENG[1,2] & Zongben XU[1]

[1]*School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China*
[2]*Pengcheng Laboratory, Shenzhen 518053, China*
[3]*School of Mathematical Sciences and School of AI, Shanghai Jiao Tong University, Shanghai 200240, China*

Received 6 October 2024/Revised 5 April 2025/Accepted 30 May 2025/Published online 29 May 2026

Abstract Multistep time-series prediction of a high-dimensional system presents a persistent challenge across scientific domains, particularly with short-term data. Drawing inspiration from Takens' embedding theorem, the existing spatiotemporal information (STI) transformation framework transforms the spatial/association information of multiple variables into the temporal dynamics of a target variable, thereby achieving promising results on short-term high-dimensional sequence prediction. However, traditional STI transformation functions are constructed using regression, which does not consider the latent prior properties of complex nonstationary scenarios, resulting in the accumulation of errors or underutilization of historical data, thereby degrading performance in several cases. In this study, we propose a spatiotemporal diffusion (STD) framework using the Koopman operator to achieve an interpretable, accurate multistep prediction of short-term time series. One key feature of the proposed STD framework is its embedding of dynamic system knowledge and specific task knowledge in STI function approximation and target variable prediction. Validations across diverse short-term high-dimensional datasets and refinement experiments highlight the STD model's robustness and efficiency.

Keywords time series, multistep prediction, short term, Koopman operator, spatiotemporal information

Citation Su L Y, Shu J, Chen L N, et al. Spatiotemporal diffusion with Koopman operator for multistep prediction of short-term time-series. Sci China Inf Sci, 2026, 69(7): 172201, https://doi.org/10.1007/s11432-024-4836-1

1 Introduction

Accurate multistep prediction of future states from short-term sequences poses a considerable challenge spanning various scientific and engineering domains, including ecology [1, 2], biology [3, 4], meteorology [5, 6] and traffic [7]. This challenge is mainly attributed to short-term data or the inherent scarcity of high-quality data in these disciplines. There are two main types of time series prediction approaches. One is to approximate the state transfer function. For example, classical statistical models [8–10] and contemporary deep learning approaches [11–15] explicitly assume or learn the dynamics that govern sequence generation within state spaces. The other is to approximate the transfer function in a delay space. For example, empirical dynamic methods [16], such as S-map [12, 17], approximate delay coordinate mappings to make predictions. However, these approaches often require access to future covariate information, which is frequently unavailable in real-world applications. In addition, these approaches face challenges in providing effective predictions solely based on short-term sequences due to insufficient learning information.

Addressing the challenging task of multistep prediction from short-term sequences entails confronting limitations arising from limited time-series information and inherent nonlinear behaviors [18, 19]. Incorporating additional relevant historical variables, such as spatial data, becomes crucial to compensate for data deficiencies and ensure reliable prediction [20].

Takens' embedding theorem [21] and its generalization [22] have laid the foundation for transforming observed high-dimensional spatial information into a target variable, i.e., its prediction. Notably, Ye and Sugihara [20] introduced multiview embedding (MVE), demonstrating the effectiveness of multiple short-term time series in prediction. However, as data dimensionality increases, model selection becomes challenging, limiting applicability. Further, MVE's iterative prediction approach requires spatial information from preceding steps, thereby limiting adaptability. The spatiotemporal information (STI) transformation framework [23–28], rooted in Takens' embedding

* Corresponding author (email: junshu@mail.xjtu.edu.cn)

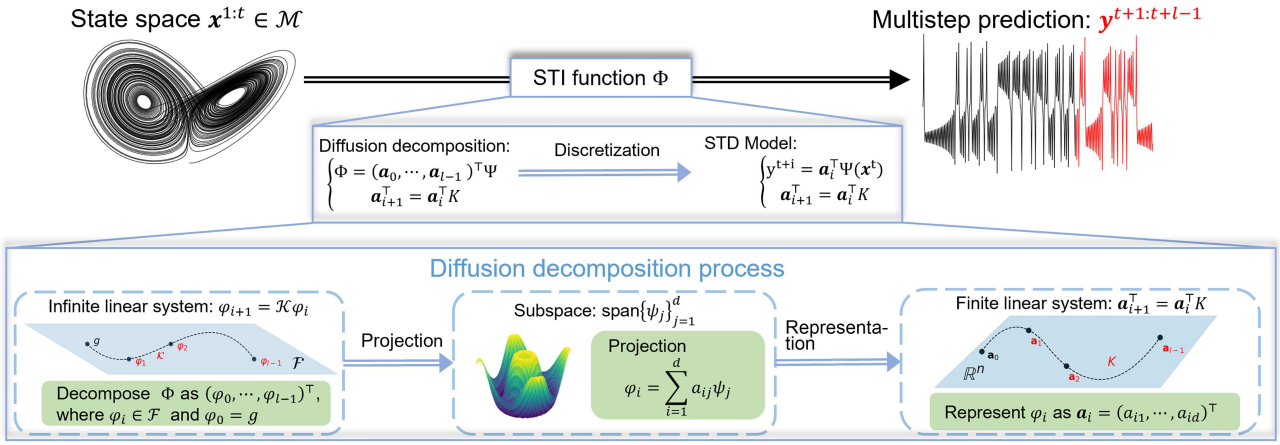

Figure 1 (Color online) STD framework. The core of the STD framework is the diffusion decomposition of the STI function, and the STD model can be seen as the discretization. The diffusion decomposition process can be divided into three steps: decomposition, projection, and representation.

theorem, represents a new way to transform embedded correlation information from observed high-dimensional data into future time series of a target variable, which not only leverages high-dimensional information but also avoids the MVE's dimensionality explosion. Particularly, the RDE method [24] within the STI framework formulates transformation functions via the bagging algorithm [29], enabling one-step prediction from short-term sequences. By incorporating reservoir computing [30] into the STI framework, the ARNN method [28] has improved prediction performance in several scenarios [31].

Despite the advancements of the STI framework, several practical issues remain. On the one hand, previous studies have empirically validated the existence of the transformation function, and practitioners often heuristically predefine certain function classes to construct transformation functions. They have limited considerations on the STI transformation function characteristics for short-term prediction tasks. Therefore, the STI framework suffers from generalization degradation due to limited samples and substantial noise in real-world applications. On the other hand, earlier methods use heuristic multi-stage algorithms to process known data in batches, resulting in information loss. This problem restricts accurate predictions in practice.

In this study, we make the following major contributions. (i) We propose a spatiotemporal diffusion (STD) model (as shown in Figure 1) for multistep predictions based on Koopman operator theory [32, 33] to characterize the intrinsic properties of STI. (ii) We present a semi-supervised regression optimization framework to solve the STD model and introduce some specific constraints based on data features to improve prediction accuracy and robustness. Particularly, the proposed model offers a unified understanding of previous STI-based heuristic algorithms, naturally leading to superior accuracy compared to previous state-of-the-art models. (iii) A variable projection algorithm is used to solve the STD model, ensuring the solution's convergence and optimality. Comprehensive analyses of three pertinent real datasets confirm the robustness and reliability of the STD model. (iv) Besides its prediction capabilities, the STD model can function as a general post-processing tool to refine other time-series models to effectively improve multistep prediction performance. The refinement experimental results of multiple models demonstrate the rationality and potential usefulness of the proposed STD model.

2 Background

2.1 Notations

Throughout, matrices (such as $X \in \mathbb{R}^{d\times m}$) are written in uppercase, vectors ($\mathbf{x} \in \mathbb{R}^d$) are written in bold lowercase, and scalars ($x \in \mathbb{R}$) are written in plain lowercase. $X_{i,\cdot}$ denotes the ith row of X and $X_{\cdot,j}$ denotes the jth column of X. $\mathbf{x}_{i:j}$ denotes $(x_i, \cdots, x_j)^\top \in \mathbb{R}^{j-i}$ where $i < j$. The Hadamard product is denoted by the symbol $\odot$. The inner product of two matrices $X \in \mathbb{R}^{k\times m}, Y \in \mathbb{R}^{k\times n}$ is denoted by $\langle X, Y\rangle = \mathrm{tr}(X^\top Y)$. For a matrix X, the maximum eigenvalue is denoted by $\lambda_{\max}(X)$, $\|X\|_F$ denotes the Frobenius norm of X, and $\|X\|$ denotes the spectral norm of X. The commonly used notations and symbols are listed in Table 1.

We adopt the root mean square error (RMSE), mean absolute error (MAE), and Pearson correlation coefficient

Table 1 Commonly used notations and symbols.

Notation	Definition	Notation	Definition	Notation	Definition
$\mathbf{x}^t$	State at time t	y^t	Target variable at time t	$\mathbf{y}^t$	Delay vector of y^t at time t
m	Number of samples	l	Number of delay	d	Spatial dimension
$\mathcal{F}$	Hilbert space	F	State transfer function	g	Observation function
$\mathcal{H}$	Hankel operator	$\mathcal{H}^{-1}$	Anti-Hankel operator	$\mathcal{P}$	Projection operator
Φ	STI function	φ_i	i-th step prediction funtion	$\mathbf{a}_i$	Finite representation of φ_i
$\mathcal{M}$	State space	$\mathcal{K}$	Koopman operator	K	The finite representaion of $\mathcal{K}$
λ	Eigenvalue of $\mathcal{K}$	$\lambda_{max}(A)$	Max eigenvalue of A	ψ	Eigenfunction of $\mathcal{K}$
γ	Regularization parameter	∇^2	Second-order difference operator	Ω	Known index of $\mathcal{H}(\mathbf{y})$

(PCC) as metrics to evaluate model predictive performance. These metrics can be calculated as follows:

$$\begin{aligned}
&\mathrm{RMSE} = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l-1}(\hat{y}_i - y_i)^2}, \qquad \mathrm{MAE} = \frac{1}{l-1}\sum_{i=1}^{l-1}|\hat{y}_i - y_i|,\\
&\mathrm{PCC} = \frac{\sum_{i=1}^{l-1}\left[(\hat{y}_i - \mathbb{E}(\hat{\mathbf{y}}))(y_i - \mathbb{E}(\mathbf{y}))\right]}{\sqrt{\sum_{i=1}^{l-1}\left(\hat{y}_i - \mathbb{E}(\hat{\mathbf{y}})\right)^2}\sqrt{\sum_{i=1}^{l-1}\left(y_i - \mathbb{E}(\mathbf{y})\right)^2}},
\end{aligned} \tag{1}$$

where $l-1$ is the prediction length, $y_i, \hat{y}_i \in \mathbb{R}$ denote the ground truth and prediction results at time i, respectively, and $\mathbb{E}(\mathbf{y}) = \frac{1}{l-1}\sum_{i=1}^{l-1} y_i$ denotes the mean value of sequence $\mathbf{y} = \{y_i\}_{i=1}^{l-1}$. The proportionate reduction of error (PRE) is used for the refinement experiments, which is calculated as follows:

$$\mathrm{PRE} = \frac{E_{\mathrm{base}} - E_{+\mathrm{STD}}}{E_{\mathrm{base}}}, \tag{2}$$

where E_{base} is the error of a base model, and $E_{+\mathrm{STD}}$ is the error of the base model refined by the STD model.

2.2 STI transformation

In short-term high-dimensional prediction tasks, the construction of STI transformation model [24,28,31] is crucial for capturing the relationship between spatial and temporal information. When observing a d-dimensional spatial state $\mathbf{x}^t = (x_1^t, x_2^t, \cdots, x_d^t)^\top \in \mathbb{R}^d$ at each time instance t, a delay vector $\mathbf{y}^t = (y^t, y^{t+1}, \cdots, y^{t+l-1})^\top \in \mathbb{R}^l$ of an arbitrary target variable $y^t = g(\mathbf{x}^t)$ can be constructed. Here, $g(\cdot)$ denotes an observation function mapping from the state space $\mathcal{M}$ to $\mathbb{R}$, and the symbol $\top$ denotes vector transposition. According to Takens' embedding theorem [21,22], if l is sufficiently large, then $\mathbf{y}^t$ can be topologically equivalent to $\mathbf{x}^t$. Thus, $\mathbf{x}^t$ has a one-to-one correspondence with $\mathbf{y}^t$. Therefore, the STI transformation equation [24] is formalized as follows:

$$\Phi(\mathbf{x}^t) = \mathbf{y}^t, \tag{3}$$

where $\Phi : \mathbb{R}^d \mapsto \mathbb{R}^l$ is a differentiable function.

2.3 Koopman operator method

Koopman theory offers a new way to analyze, predict, and control nonlinear systems in the past decade [34–40], in which nonlinear dynamics can be represented by an infinite-dimensional linear operator acting on the space of all possible system observation functions. Suppose that the collected sequential data are sampled from an unknown discrete-time dynamical system in a state space $\mathcal{M}$ of the form

$$\mathbf{x}^{t+1} = F(\mathbf{x}^t), \tag{4}$$

where $\mathbf{x}^t \in \mathcal{M} \subseteq \mathbb{R}^d$ is the system state at time t and $F : \mathcal{M} \mapsto \mathcal{M}$ is a vector field describing the system dynamics. Let $\mathcal{F}$ be a Hilbert space of real-valued functions on $\mathcal{M}$. For any scalar observable function, $g \in \mathcal{F} : \mathcal{M} \mapsto \mathbb{R}$, the Koopman operator $\mathcal{K}$ acts on it by composition with the following dynamics:

$$\mathcal{K}g = g \circ F. \tag{5}$$

The Koopman operator $\mathcal{K}$ is clearly a linear operator. According to the spectral theory of the linear operator [41], if $\mathcal{K}$ has an eigenfunction $\psi(x)$ corresponding to an eigenvalue λ satisfying

$$\psi(\mathbf{x}^{t+1}) = \mathcal{K}\psi(\mathbf{x}^t) = \lambda\psi(\mathbf{x}^t), \tag{6}$$

then for a given observation $g \in \mathcal{F}$, if there exists a set of Koopman eigenfunctions $\{\psi_j\}_{j=1}^{\infty}$ that form a Koopman invariant subspace $\text{span}\{\psi_j\}_{j=1}^{\infty}$ of observation g, i.e.,

$$g = \sum_{j=1}^{\infty} a_j\psi_j \Rightarrow \mathcal{K}^t g = \sum_{j=1}^{\infty} a_j\lambda_j^t\psi_j. \tag{7}$$

The dynamics of the observable g can be completely linearly characterized in the invariant subspace $\text{span}\{\psi_j\}_{j=1}^{\infty}$. In practice, it is possible to approximate this expansion as a truncated sum of only a few dominant terms [40, 42].

3 Spatiotemporal diffusion model

Considering the dynamical system (4), for any bounded observable function $g \in \mathcal{F}$ and $i \geqslant 0$, the future target of step i forward can be expressed as follows:

$$y^{t+i} = g \circ F^i(\mathbf{x}^t) = g(\mathbf{x}^{t+i}), \tag{8}$$

where F^i denotes the composition of F i times. Usually, in practical prediction, the future spatial variable $\mathbf{x}^{t+i}$ is typically unattainable in advance: the prediction can only be made based on current and historical information $\{\mathbf{x}^{t-j} \mid 0 \leqslant j \leqslant t-1\}$. Therefore, exploring historical observations for future prediction is more consistent with realistic prediction scenarios than focusing on the complex nonlinear dynamics of high-dimensional state variables. By combining (5) based on the Koopman operator property, the observation from the historical state space to the future target can be represented as follows:

$$y^{t+i} = (\mathcal{K}^i g)(\mathbf{x}^t). \tag{9}$$

For brevity, we define the observation function of the i-th step forward $\mathcal{K}^i g$ as φ_i and $\varphi_0 = g$. The observation functions obey the following infinite-dimensional linear dynamical system:

$$\varphi_i = \mathcal{K}^{i-j}\varphi_j, i \geqslant j \geqslant 0. \tag{10}$$

Based on this understanding, the STI transformation function in (3) can be decomposed as a set of states in the infinite-dimensional linear dynamical system:

$$\begin{aligned} \Phi &= (\mathcal{K}^0, \mathcal{K}^1, \cdots, \mathcal{K}^{l-1})^\top g \\ &= (\varphi_0, \varphi_1, \cdots, \varphi_{l-1})^\top. \end{aligned} \tag{11}$$

We call the above decomposition the diffusion decomposition of the STI transformation function. It offers a methodology for constructing Φ by diffusing the observation g along time t. Notably, the decomposition depends on two factors: the observation function g and the linear diffusion operator $\mathcal{K}$. On the one hand, g, although known, may exhibit diverse properties based on the task at hand. Certain properties should be upheld during the short-term diffusion process due to the system's temporal consistency. On the other hand, according to the spectral decomposition of the Koopman operator, if there exists a finite set of basis functions $\Psi = (\psi_1, \psi_2, \cdots, \psi_n)^\top$ that approximate a Koopman invariant subspace of observations $\{\varphi_i\}_{i=0}^{l-1}$ [40], then each observation can be formed as linear combinations of the finite basis functions as follows:

$$\varphi_i = \mathbf{a}_i^\top \Psi, i \in \{0, 1, \cdots, l-1\}, \tag{12}$$

where $\mathbf{a}_i \in \mathbb{R}^n$ is a n-dimensional representation of the observation function φ_i, which can be used for numerical computation. According to the spectral decomposition theory of the Koopman operator, the dynamics of this finite-dimensional representation, $\mathbf{a}_i$, obey the following linear dynamics:

$$\mathbf{a}_{i+1}^\top = \mathbf{a}_i^\top K, \tag{13}$$

where $K \in \mathbb{R}^{n\times n}$. Therefore, the linear dynamics provide a comprehensive understanding of the dynamical behavior of the observation functions $\{\varphi_i\}_{i=0}^{l-1}$. In summary, the proposed STD model for multistep predictions can be expressed as follows:

$$\begin{cases} y^{t+i} = \mathbf{a}_i^\top \Psi(\mathbf{x}^t), \\ \mathbf{a}_{i+1}^\top = \mathbf{a}_i^\top K, \end{cases} i = 0, 1, \cdots, l-1. \tag{14}$$

Notably, the aforementioned model can make predictions by establishing relationships between historical observations and future predictions. Compared with the traditional iterative prediction model, the proposed model can formulate relationships between different observations, effectively avoiding the accumulation of errors. In addition, the proposed model provides a more robust theoretical foundation and stronger interpretability than the general multistep prediction model. This is substantiated by the results of experiments conducted on different datasets, as illustrated in the following section.

3.1 Semi-supervised regression objective

In practice, given a length-m d-dimensional trajectory $X = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^m\} \in \mathbb{R}^{d\times m}$, sampled from an unknown dynamical system, a target variable y^t, e.g., x_k^t, is observed. This study aims to predict the target variable $y^{m+1:m+l-1}$ in $l-1$ steps solely using the past information of X. To make multistep prediction, the STI framework (3) needs an algorithm $\mathcal{A} : \{\mathbf{x}^t, \mathbf{y}^t\}_{t=1}^m \mapsto \Phi$. Notably, $\mathbf{y}^t$ is partially known if $t > m-l$, so the data pair $\{\mathbf{x}^t, \mathbf{y}^t\}_{t=1}^m$ is partially known and the learning task $\mathcal{A} : \{\mathbf{x}^t, \mathbf{y}^t\}_{t=1}^m \mapsto \Phi$ is a semi-supervised task [43]. When solving the above problems, previous algorithms [24, 28] based on STI employ a two-stage approach that (1) uses part of the known data $\{\mathbf{x}^t, \mathbf{y}^t\}_{t=1}^{m-l}$ to solve a temporary solution Φ_0 by supervised learning and (2) iteratively completes $\mathbf{y}^t$ by $\Phi_0(\mathbf{x}^t) = \mathbf{y}^t, t > m-l$ and converts the task into a supervised task to solve Φ. In short, Φ is solved after obtaining the labels with errors based on partial information. These algorithms fail to exploit the data information and exhibit information loss, resulting in error in the obtained solutions.

We are now in a position to demonstrate an important result about the solution to the STD model (14): given a linear invariant system with complete data, the optimal solution that minimizes the RMSE is consistent with the linear dynamic property expressed in (13). The following linear system experiment confirms this proposition.

Proposition 1. Suppose that for some state space $\mathcal{M} \subset \mathbb{R}^n$, for $t \in [0, T]$, the system satisfies the linear dynamic $\mathbf{z}^{t+1} = K\mathbf{z}^t$ in the state space, $\mathbf{z}^t \in \mathcal{M}\backslash\{\mathbf{0}\}$, where $K \in \mathbb{R}^{n\times n}$ is non-singular and with n distinct eigenvalues, and $y^t = \mathbf{b}^\top \mathbf{z}^t$ is a linear observation from the system, where $\mathbf{b} \in \mathbb{R}^n$. For any $i \in \{0, 1, \cdots, T-n\}$ and $n \leqslant \tau_i < T-i$, let

$$\mathbf{a}_i^* = \arg\min_{\mathbf{a}_i} \sum_{t=0}^{\tau_i} \left(\mathbf{a}_i^\top \mathbf{z}^t - y^{t+i}\right)^2. \tag{15}$$

Suppose further that $\mathbf{z}^0 = \sum_{i=1}^n c_i \mathbf{w}_i$, where $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ are eigenvectors of K and $c_1, c_2, \ldots, c_n \neq 0$. Then for any $i \in \{1, 2\cdots, T\}$, the solution satisfies $\mathbf{a}_i^* = K^\top \mathbf{a}_{i-1}^*$.

According to Proposition 1, it is possible to merge the linear dynamics constraint (13) into the error term, thereby formulating the STD model (14) into a general optimization problem.

$$\begin{aligned} \min_{A,\hat{\mathbf{y}}} \quad & \|A\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})\|_F^2 + \gamma\mathcal{R}(A), \\ \text{s.t.} \quad & \|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\| < \epsilon, \end{aligned} \tag{16}$$

where $\|\cdot\|_F$ is the Frobenius norm and ϵ denotes the level of noise present in the data which is often present in real-world data, $\mathcal{R}(\cdot)$ is other prior information for A which can be designed according to the data characteristics, with several examples presented in Subsection 3.2. $\mathcal{H}$ is a Hankel operator, which is defined for $\mathbf{y} = \{y^1, y^2, \cdots, y^m, y^{m+1}, \cdots, y^{m+l-1}\}$ as follows:

$$\mathcal{H}(\mathbf{y}) = \begin{bmatrix} y^1 & \cdots & y^{m-1} & y^m \\ \vdots & \ddots & \vdots & \vdots \\ y^{l-1} & \cdots & y^{l+m-3} & y^{l+m-2} \\ y^l & \cdots & y^{l+m-2} & y^{l+m-1} \end{bmatrix}. \tag{17}$$

Notably, previous heuristic algorithms [24, 28] based on the STI transformation equation can be unified into the above framework; the details are outlined in Appendix B. The unified optimization framework and explicit physical implications of the proposed model could potentially address the shortcomings of previous algorithms and improve performance.

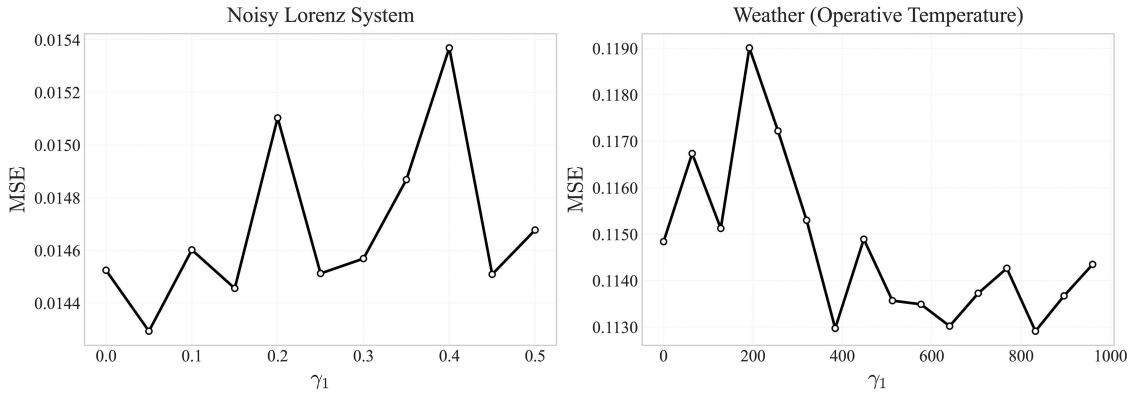

Figure 2 Ablation experiments for the smoothing term in two datasets: the Lorenz system and weather dataset.

3.2 Regularization term for A

In practice, the target variable y^t is usually chosen as one of the state variables $\mathbf{x}^t$, for example $y^t = x_k^t$, $k \in \{1, 2, \cdots, d\}$. In such a task, the corresponding observation function g is

$$g(\mathbf{x}^t) = y^t. \tag{18}$$

Observe that g can deduce the sparsity of $\mathbf{a}_0$ in (14) based on the following proposition.

Proposition 2 (Sparse linear observation)**.** Let $\mathbf{x} = (x_1, x_2, \cdots, x_d)^\top \in \mathbb{R}^d$ and $g : \mathbb{R}^d \mapsto \mathbb{R}$, if there exists $k \in \{1, 2, \cdots, d\}$ such that $g(\mathbf{x}) = x_k$, there is a unique sparse vector $\mathbf{e}_k \in \mathbb{R}^d$ such that

$$g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{e}_k \rangle, \forall \mathbf{x} \in \mathbb{R}^d. \tag{19}$$

Thus, we could set an l_1 sparse regularization [44] for A to control the prediction model, thereby enhancing the generalization and robustness for short-term high-dimensional prediction.

Typically, the target contains some high-frequency noise, and the true curve is usually locally smoother. Therefore, making the prediction smoother often improves robustness (Figure 2). We can constrain the smoothness of the curve by its second derivative [9]. Smoother curves tend to have smaller second-derivative sums, whereas steeper curves tend to have larger second-derivative sums. In practice, we can use the second-order backward differences of $\mathbf{a}_i$ to constrain the smoothness of the target curve because $\nabla^2 y_{t+i} = \nabla^2 \mathbf{a}_i \Psi(\mathbf{x}_t)$ according to the STD model (14). ∇^2 is a second-order backward difference operator defined by

$$\nabla^2 \mathbf{a}_i = \begin{cases} \mathbf{0}, & i = 1, \\ \mathbf{a}_i - \mathbf{a}_{i-1}, & i = 2, \\ \mathbf{a}_i - 2\mathbf{a}_{i-1} + \mathbf{a}_{i-2}, & i > 2. \end{cases} \tag{20}$$

In summary, the following model can be obtained by embedding the above two prior properties:

$$\begin{aligned} \min_{A, \hat{\mathbf{y}}} \quad & \|\mathbf{A}\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})\|_F^2 + \gamma_1 \|\nabla^2 \mathbf{A}\|_F^2 + \gamma_2 \|\mathbf{A}\|_1, \\ \text{s.t.} \quad & \|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\| < \epsilon, \end{aligned} \tag{21}$$

where γ_1 and γ_2 are hyperparameters that determine the strength of the prior.

As shown in Figure 2, the smoothness term is empirically effective. As the constraint strength is increased, the model performance initially increases before decreasing, suggesting the presence of high-frequency noise in the sequence. Certain smoothness constraints appear to mitigate its effect. However, there is also some useful high-frequency information in the data, and too strong a smoothness constraint will weaken the model performance.

Nevertheless, alternative regularization terms may prove more effective in other circumstances. Fortunately, the STD method often produces useful results with the above choices of regularization term.

3.3 Optimization

The variable projection method [45] can be used to solve the above problem (21). Refer to Appendix C for the details. A brief overview of the solution process is presented here. For a fixed A, we define the solution that minimizes $\|A\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})\|_F^2 + \frac{1}{\lambda}\|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\|_2^2$ as $\hat{\mathbf{y}}$.

Table 2 Runtime comparison. The runtime of the STD model with warm-up represents the median value across all test configurations.

Method	TRMF	SSA	MVE	RDE	ARNN	STD w/o warm-up	STD w/ warm-up
Run time (s)	3.305	0.060	0.021	0.003	0.442	0.419	0.061

Thus the minimization problem (21) can be rewritten in term of A alone, which can be reduced to

$$\min_{A} \|\mathcal{P}_{\Omega}(A\Psi(X) - \mathcal{H}(\hat{\mathbf{y}}))\|_F^2 + \gamma_1\|\nabla^2 A\|_F^2 + \gamma_2\|A\|_1, \tag{22}$$

where $\mathcal{P}$ is a projection operator and the index set of $\hat{\mathbf{y}}_{1:m}$-elements in $\mathcal{H}(\mathbf{y})$ is denoted by Ω. We use the fast iterative shrinkage thresholding algorithm [46] to solve the matrix optimization problem (22). This corresponds to lines 2–6 of Algorithm 1. The following proposition gives the convergence rate of the algorithm.

Proposition 3. Let $F(A) = \|\mathcal{P}_{\Omega}(A\Psi(X)-\mathcal{H}(\mathbf{y}))\|_F^2+\gamma_1\|\nabla^2 A\|_F^2+\gamma_2\|A\|_1$, $\{A_k\}, \{B_k\}$ be generated by algorithm 1. Then for any $k > 1$,

$$F(A_k) - F(A^*) \leqslant \frac{2C\|A_0 - A^*\|_F^2}{(k+1)^2}, \tag{23}$$

where $C = 2(\lambda_{\max}(\Psi(X)\Psi(X)^{\top}) + \gamma_1\lambda_{\max}(\nabla^{2^{\top}}\nabla^2))$ and $\lambda_{\max}(A)$ denotes the maximum eigenvalue of matrix A.

As described in Proposition 3, the convergence rate of the algorithm depends mainly on the choice of initial value A_0. The existence of time series correlation in the data itself is a key factor in determining the choice of initial value. A natural initial value is selected as the result of the solution in the last step. Table 2 illustrates the mean time over 50 runs required for the algorithm to converge in the Lorenz system under both warm start and cold start conditions. The warm start technique can accelerate convergence.

Upon acquiring the representation of observations A^*, the prediction for the problem (21) is $\hat{y} = \mathcal{H}^{-1}(A^*\Psi(X))$ where $\mathcal{H}^{-1}$ is an anti-Hankel operator. In practice, the prediction $\hat{\mathbf{y}}$ is determined by solving the subproblem:

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \|\mathcal{H}(\mathbf{y}) - A^*\Psi(X)\|_F^2. \tag{24}$$

This problem possesses a closed-form solution [47] achieved through anti-diagonal averaging of $A^*\Psi(X)$. We can compute the sum of each anti-diagonal of $A^*\Psi(X)$, denoted by t_n, as follows:

$$t_n = \begin{cases} n, & n = 1, \cdots, l, \\ l, & n = l+1, \cdots, m, \\ m+l-n, & n = m+1 \cdots, m+l-1. \end{cases} \tag{25}$$

Then the prediction $\hat{y}_i$ is given by

$$\hat{y}_i = \frac{1}{t_i} \sum_{p+q=i+1} [\mathbf{A}^*\Psi(X)]_{p,q}, i \in \{m+1, \cdots, m+l-1\}. \tag{26}$$

The computation complexity of the STD model is $O(dlm)$, and the memory consumption is $O(dl)$. A detailed analysis is given in Appendix C.

Algorithm 1 STD model.

Require: Given dataset $X \in \mathbb{R}^{d\times m}$, $\mathcal{H}(\mathbf{y}) \in \mathbb{R}^{l\times m}$, index set Ω;
1: Initialize A_0, B_0, $a_0 = 1$ and step size $\tau = \frac{1}{2(\lambda_{\max}(\Psi(X)\Psi(X)^{\top})+\gamma_1\lambda_{\max}(\nabla^{2^{\top}}\nabla^2))}$;
2: **repeat**
3: $P_{t+1} = \gamma_1\nabla^{\top}\nabla A_t + \mathcal{P}_{\Omega}(A_t\Psi(X) - \mathcal{H}(\mathbf{y}))\Psi(X)^{\top}$;
4: $A_{t+1} = \mathrm{Prox}_{\tau\gamma_2}(B_t - 2\tau P_{t+1})$;
5: $a_{t+1} = \frac{1+\sqrt{1+4a_t^2}}{2}$;
6: $B_{t+1} = A_t + \frac{a_t-1}{a_{t+1}}(A_{t+1} - A_t)$;
7: **until** convergence
8: $\hat{\mathbf{y}} = \mathcal{H}^{-1}(A_{t+1}\Psi(X))$;
Ensure: $\hat{\mathbf{y}}$;

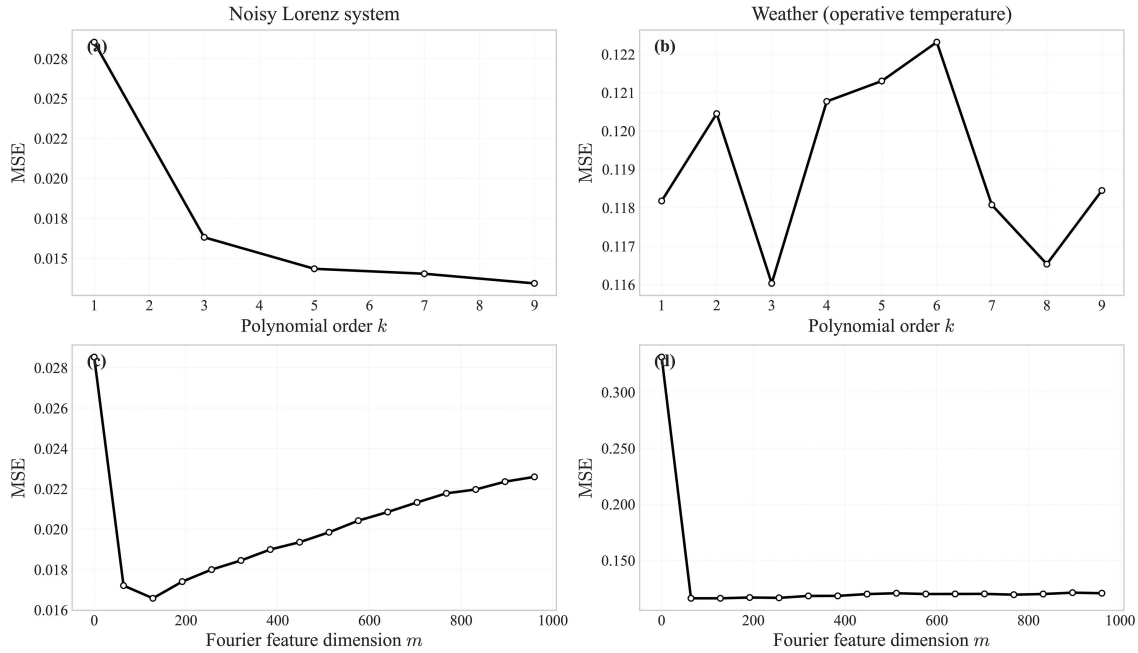

Figure 3 Ablation results for polynomial basis and RFF in two datasets: the Lorenz system and weather dataset. The model performance first increases and then decreases as the polynomial order or RFF dimension increases, indicating that too many dimensions may result in overfitting and reduce test performance.

3.4 The choice of Ψ

In real-world systems, the optimal choice of basis functions to represent dynamics may not be evident. Although the optimal choice of Ψ is unknown, there are some choices in estimating the Koopman invariant subspace [42]: a manual dictionary [42], kernel methods [48], or a neural network [49]. Training a neural network to consistently and robustly estimate the Koopman invariant subspace is difficult and often requires a large amount of data. However, the domain on which the underlying dynamical system is defined is not always known, or is often only partially observed in practice. The Legendre polynomials and random Fourier feature (RFF) [50] are employed to approximate the Koopman invariant subspace. The Legendre polynomials are simple to implement and are conceptually related to the approximation of the Koopman eigenfunctions with a Taylor expansion. The RFF is an effective method to approximate a stationary kernel; a d-dimensional RFF is defined as follows:

$$\Psi_{RFF}(x) = \sqrt{\frac{2}{d}}\left[\cos(\omega_1 x + b_1), \cdots, \cos(\omega_d x + b_d)\right], \omega_i \sim \mathcal{N}(0,1), b_i \sim \mathcal{U}(0, 2\pi). \tag{27}$$

Figure 3 shows the ablation results. The model performance first increases and then decreases as the polynomial order or RFF dimension increases. This observation implies that too many dimensions may result in overfitting and tend to impair test performance; thus third-order polynomials or RFF with 128 dimensions are used in our implementation.

Other Ψ may be more effective in other circumstances. How to choose the best set of Ψ is an important, yet open question; fortunately, the STD method often produces useful results even with the relatively naive choices of Ψ presented in this section.

The STD model validation includes robustness experiments on simulation data, prediction experiments on multiple real-world datasets, and its application to enhance the multistep prediction of classical time series and previous based on the STI-based models. The results confirm the STD model's effectiveness in integrating multiple types of information, as well as enhancing the robustness and accuracy of multistep prediction.

4 Demonstration on examples

All experiments are implemented in Pytorch [51]. All the baselines that we reproduced are implemented based on the default or original paper configurations. The parameter settings for all experiments are presented in Appendix G.

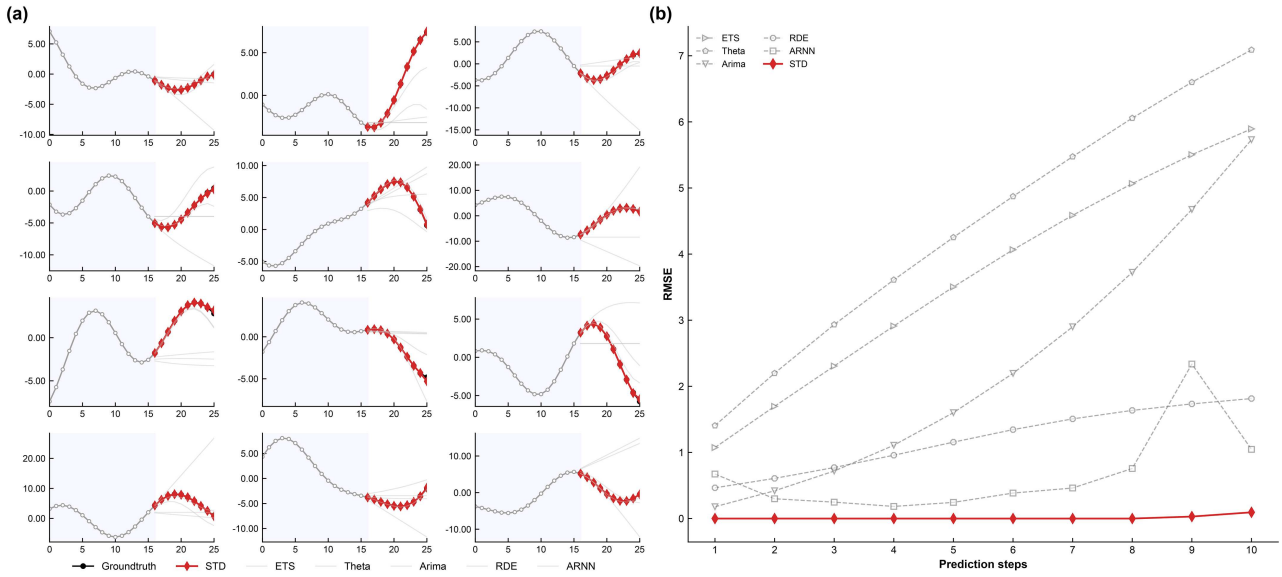

Figure 4 (Color online) Comparison of prediction results of different methods on a linear system. (a) Different plots represent prediction results of different methods on test samples. The x-axis denotes the time steps, while the y-axis represents the sequence values. The proposed STD model outperforms the other models in all test data. (b) RMSE of the prediction results across different prediction steps in a linear system. The prediction performance of other models deteriorates with an increase in the prediction step size, but the STD model shows its robustness for linear systems.

We compare the well-acknowledged models in experiments, including statistical models: ARIMA [10, 52], ETS [8,52], and Theta [9,52]; dynamical system models: ARNN [28], RDE [24], and MVE [20]; matrix estimate methods: SSA [53], TRMF [54], and neural network method: iTransformer [13], SegRNN [14], and NBeats [15]. As the experiments involved multistep predictions, the traditional methods are limited to making linear predictions, and, in some cases, the PCC is not applicable. Consequently, the subsequent results exclude the PCC results of the traditional methods.

4.1 Multistep prediction experiments

4.1.1 *A linear system case*

In the first experiment, a simple 8D linear system is used to demonstrate the STD model's effectiveness. We varied the number of prediction horizons, fixed the training data at 16, and evaluated the performance of different models. A prediction of 10 steps was made with an input set containing 16 step data (Figure 4(a)). The proposed STD model was validated to make more precise predictions about linear systems on different prediction horizons than the traditional models and previous heuristic models (Figure 4(b)).

This experiment confirms that the semi-supervised regression objective and the prior property of the prediction model improve multistep prediction compared with previous heuristic models.

4.1.2 *Performance on the Lorenz system*

Besides its efficacy in linear systems, the STD model demonstrates comparable performance in nonlinear systems, the Lorenz system. STD consistently outperforms other models in two cases: forecasting the Lorenz system with different noise levels and input lengths. Tables 3 and 4 summarize the prediction results of seven models in both cases. The results demonstrate that STD is robust and adaptive to diverse challenging circumstances.

For each case, multiple experiments were conducted, with the initial points randomly sampled. The RMSE and PCC of the averaged predictions were then recorded. In the first experiment, we varied the amount of training data provided to the models and kept the noise variance fixed at 0.5. Each model was provided with 15, 18, 21, 24 and 27 samples of $\mathbf{x}_t$, and the performance of predicting the next 12 snapshots is evaluated. Notably, because the prediction length must be less than half of the input length for ARNN, it is excluded from this comparison. In the next experiment, we probed the robustness of the respective models to noise by keeping the amount of training data fixed ($m = 27$) and varying the variance of the Gaussian noise corrupting the signal. Figure 5 shows the results.

Notably, Figure 5 indicates that the STD method achieves relatively better robustness against data noise than other models, effectively capturing the essential data structure. Its prediction results were substantially improved

Table 3 Robustness of STD to observational noise. The table indicates that STD outperforms other models on the Lorenz system with noise. The best candidates are in bold, and the second-best candidates are underlined.

Method	0.0			0.2			0.5			0.7			1.0		
	PCC	MAE	RMSE	PCC	MAE	RMSE	PCC	MAE	RMSE	PCC	MAE	RMSE	PCC	MAE	RMSE
Arima	0.433	0.061	0.080	–	0.127	0.149	–	0.095	0.106	–	0.088	0.096	–	0.089	0.095
ETS	–	0.101	0.119	–	0.114	0.131	–	0.117	0.132	–	0.117	0.130	–	0.111	0.123
Theta	–	0.120	0.136	–	0.119	0.134	–	0.125	0.137	–	0.125	0.136	–	0.123	0.133
SSA	0.038	0.110	0.129	0.192	0.087	0.096	0.193	0.080	0.089	0.193	0.075	0.082	0.192	0.066	0.073
TRMF	0.554	0.089	0.100	0.698	0.091	0.102	0.606	0.090	0.100	0.619	0.081	0.089	0.519	0.074	0.081
ARNN	0.937	0.026	0.031	0.935	0.034	0.039	0.808	0.061	0.074	0.806	0.043	0.051	0.669	0.049	0.059
MVE	0.904	0.020	0.026	0.883	0.022	0.028	0.770	0.029	0.036	0.701	0.034	0.041	0.635	0.039	0.047
RDE	**0.995**	0.016	0.018	**0.993**	0.017	0.018	**0.984**	0.019	0.021	**0.970**	0.022	0.024	0.924	0.024	0.027
STD	**0.995**	**0.011**	**0.012**	0.986	**0.010**	**0.011**	0.973	**0.014**	**0.015**	0.956	**0.016**	**0.018**	**0.925**	**0.017**	**0.020**

Table 4 Robustness to input length. The table indicates that STD outperforms other models on the Lorenz system with noise. The best candidates are in bold, while the second-best candidates are underlined.

Method	15			18			21			24			27		
	PCC	MAE	RMSE	PCC	MAE	RMSE	PCC	MAE	RMSE	PCC	MAE	RMSE	PCC	MAE	RMSE
Arima	–	0.117	0.128	–	0.109	0.119	–	0.105	0.115	–	0.107	0.119	–	0.095	0.106
ETS	–	0.129	0.145	–	0.121	0.133	–	0.102	0.114	–	0.100	0.111	–	0.117	0.132
Theta	–	0.136	0.155	–	0.148	0.165	–	0.136	0.151	–	0.135	0.149	–	0.125	0.138
SSA	0.013	0.129	0.139	0.127	0.114	0.121	0.159	0.099	0.110	0.099	0.087	0.098	0.193	0.080	0.089
TRMF	0.349	0.112	0.121	0.576	0.091	0.100	0.577	0.083	0.092	0.547	0.083	0.092	0.606	0.090	0.100
MVE	0.549	0.084	0.096	0.647	0.075	0.085	0.684	0.057	0.068	0.766	0.038	0.047	0.770	0.029	0.036
RDE	0.367	0.124	0.133	0.680	0.094	0.100	0.867	0.066	0.070	0.960	0.040	0.043	**0.984**	0.019	0.021
STD	**0.591**	**0.069**	**0.080**	**0.930**	**0.039**	**0.043**	**0.957**	**0.026**	**0.029**	**0.975**	**0.018**	**0.021**	0.973	**0.014**	**0.015**

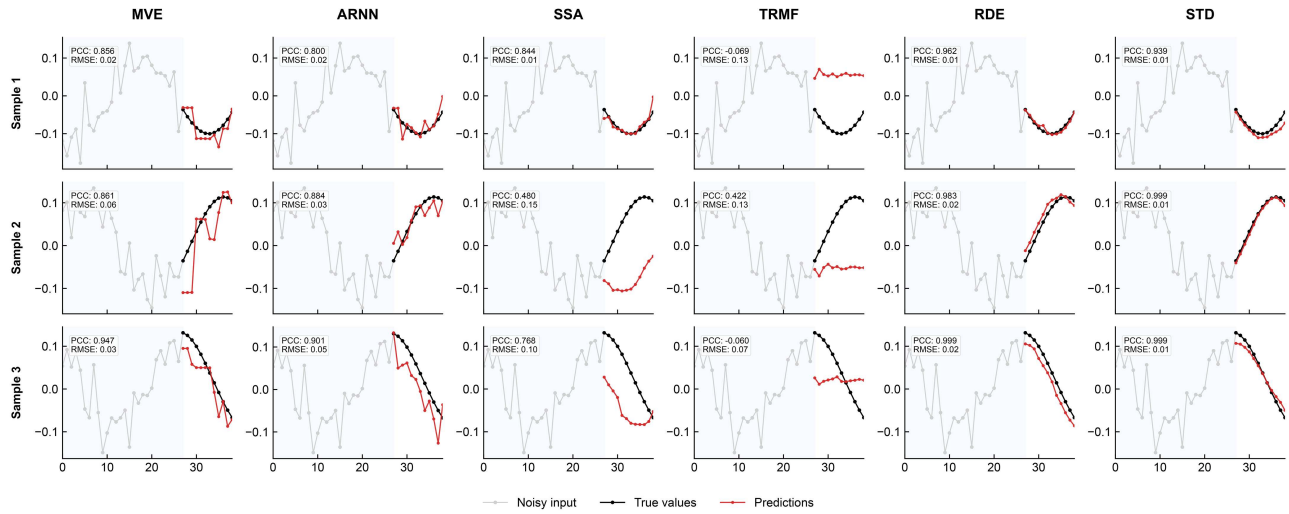

Figure 5 (Color online) Comparison of the prediction results on the Lorenz system. Each column corresponds to a model's prediction on different test samples. The x-axis denotes the time steps, while the y-axis represents the sequence values. The prediction results are shown in the top left corner of each plot. The STD model outperforms other models and is more robust to noise.

over those of other models.

For testing the deep learning methods, we generated a test set with the initial condition $\hat{\mathbf{x}}^{t_2}$ and generated a train set from t_0 to t_1 and a validation set from t_1 to t_2. The last validation point was marked as $\mathbf{x}^{t_2}$. We define the distance to measure the discrepancy between the test and historical data as follows:

$$d(\hat{\mathbf{x}}^{t_2}, \mathbf{x}^{t_2}) = \|\hat{\mathbf{x}}^{t_2} - \mathbf{x}^{t_2}\|_2^2 / \|\hat{\mathbf{x}}^{t_2}\|_2^2, \tag{28}$$

where $\|\cdot\|_2$ denotes the 2-norm.

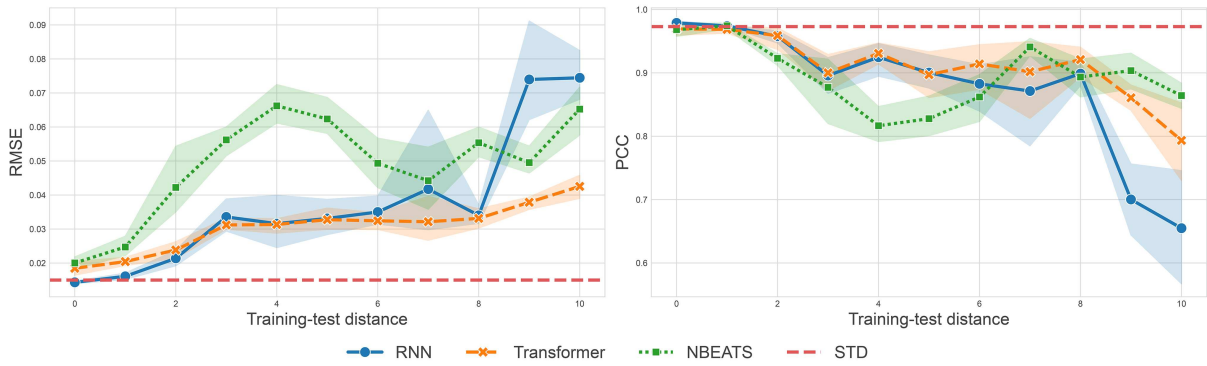

Figure 6 (Color online) Performance comparison between deep learning methods and STD as the difference between the training and test data increases on the Lorenz system.

Figure 6 shows the performance of deep learning methods with increasing distance (28). These methods depend on a large amount of historical data for training. In practice, the domain on which the underlying dynamical system is defined is not always known, or is often only partially observed. A discrepancy between the test and historical data can result in substantial performance degradation. As shown in Figure 6, the performance of several common deep learning methods [13–15] degrades to some extent as the deviation between the training and the test data increases. Meanwhile, STD is not constrained by large historical data and is capable of making inferences based on current data in real time, as shown by the red dashed lines in Figure 6.

4.1.3 *Performance on real-world datasets*

The STD model can also be applied to general real-world datasets, demonstrating its versatility and robust predictive capabilities. We demonstrate its effectiveness across multiple domains: plankton community [55], operative temperature[1)] and wind speed [56]. The proposed model's prediction results, as well as the comparisons, are summarized in Figure 5 and Table 5. Making predictions is difficult because these datasets exhibit complex interconnected dynamics without significant periodicity or trend.

Climate data are recorded every 10 min for the second half of 2020, which contains 16 meteorological indicators, including air temperature, and humidity. A 2-hour historical dataset (12 points) was used to predict the composite index operative temperature value for the next one hour (6 points). The efficacy of STD in making credible and accurate predictions is evident even when data exhibit pronounced irregular oscillations, as shown in the Figure 7(a).

The ecosystem is an important scenario for prediction based on short-term data in which datasets are often cross-sectionally wide (e.g., census of several interacting species) but short in the time dimension [57]. We consider the plankton community, including 12 species to predict 4 steps/points with a training set containing 10 points. The overall data exhibit a relatively smooth pattern, and STD yields accurate predictions compared with baseline methods as shown in Figures 7(b) and (c).

The wind speed dataset meticulously documented by the Japan Meteorological Agency, is collected at 10-minute intervals from 154 regions in Japan. The system shows a high-dimensional nonlinear property, and the 2-hour (12 points) predictions of Osaka and Fukushima are made by STD with an input set containing 24-hour (144 points) data. The nonlinear nature of wind speed data, characterized by numerous high-frequency fluctuations as shown in Figure 7(e), inherently complicates prediction endeavors. However, STD can discern and capture fundamental low-frequency trends, enabling multi-step predictions solely based on short-term data.

Figure 7 indicated that STD can make reliable multistep predictions without notable periods and trends and in the presence of several random perturbations, even in short-term data scenarios. The model's performance across diverse real-world datasets demonstrated its effectiveness in robust prediction from short-term high-dimensional data.

4.2 Refinement experiment

Previous STI-based heuristic algorithms can be unified into the STD framework, as detailed in Appendix G of the SI. The explicit theoretical foundation of STD allows for a direct improvement of the effectiveness of previous STI-based heuristic models. In addition, although classical time series models exhibit high efficiency and accuracy across

1) https://www.bgc-jena.mpg.de/wetter/.

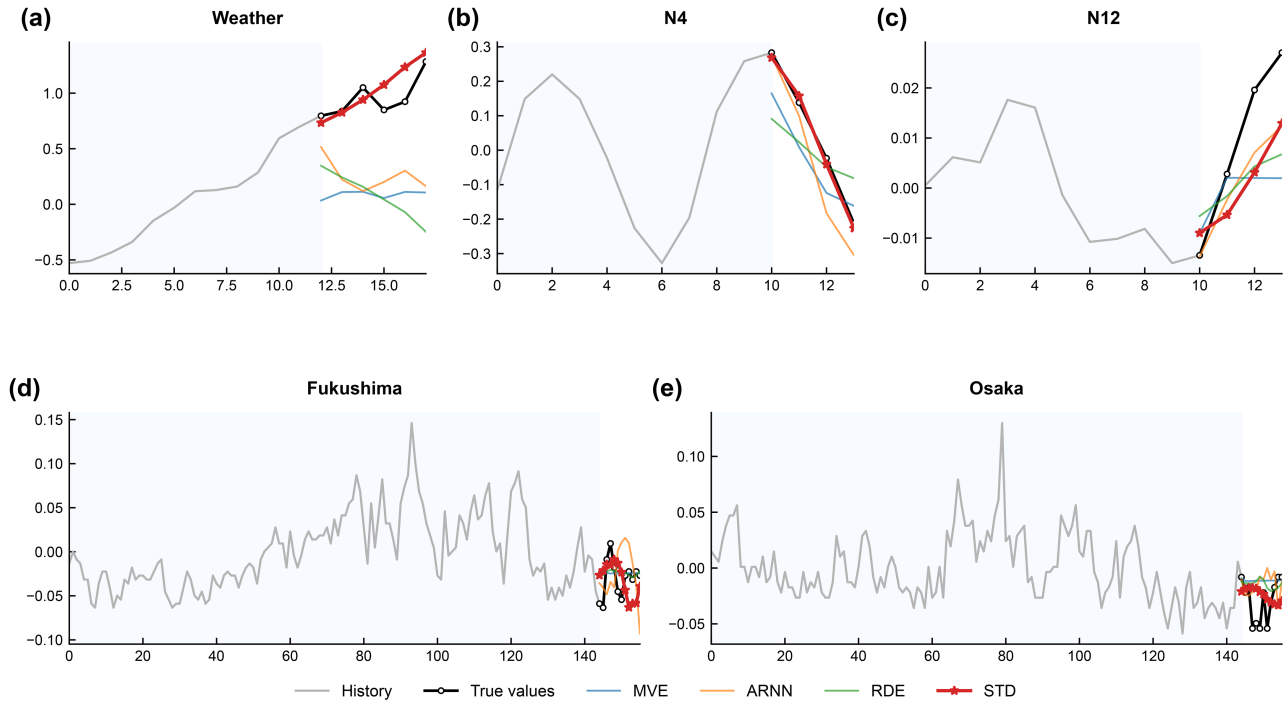

Figure 7 (Color online) Comparison of prediction results on real-world datasets. The x-axis denotes the time steps, while the y-axis represents the sequence values.

Table 5 Main empirical results on real-world datasets. Metrics are averaged over multiple runs; the best results are in bold, and the second-best results are underlined.

Method	Plankton				Wind				Weather	
	N4		N12		Osaka		Fukushima			
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Arima	0.217	0.192	0.012	0.011	0.030	0.026	0.028	0.025	0.115	0.099
ETS	0.240	0.219	0.015	0.014	0.031	0.026	0.028	0.024	0.131	0.114
Theta	0.282	0.251	0.016	0.014	0.031	0.026	0.028	0.024	0.127	0.118
SSA	0.222	0.202	0.020	0.017	0.035	0.031	0.042	0.038	0.423	0.382
TRMF	0.247	0.228	0.032	0.030	0.028	0.024	0.033	0.029	0.205	0.191
RNN	0.110	0.095	0.012	0.011	0.029	0.024	**0.025**	**0.021**	0.119	0.104
Transformer	0.161	0.141	0.013	0.012	0.029	0.025	0.027	0.023	0.223	0.210
Nbeats	**0.101**	**0.085**	0.013	0.012	0.039	0.033	0.034	0.029	0.120	0.106
ARNN	0.203	0.175	0.012	0.011	0.036	0.030	0.036	0.031	0.231	0.200
MVE	0.143	0.133	0.010	0.009	0.032	0.027	0.029	0.024	0.121	0.107
RDE	0.217	0.200	0.014	0.013	0.026	0.021	0.027	0.023	0.255	0.236
STD	0.143	0.124	**0.009**	**0.008**	**0.025**	**0.020**	0.027	0.023	**0.112**	**0.096**

real-world scenarios with long-term data [58], their reliance on iterative prediction methods for multistep forecasts and the lack of effective methods to combine multivariate information often result in error accumulation. The STD model can also alleviate this problem of the classical models or enhance their performance as a post-processing tool.

Experiments conducted on synthetic and real datasets demonstrate the effectiveness of STD in improving the performance of other models. Table 6 shows that STD consistently enhances prediction accuracy and reduces errors up to 17% in various multistep prediction scenarios. The complete refinement figures are presented in Appendix G.

5 Discussion and conclusion

Constructing an STI transformation model is crucial to make accurate predictions in short-term high-dimensional prediction scenarios. In this study, as a key contribution, we propose an interpretable STD model based on the

Table 6 Refinement results. The table displays the PRE of the various models after STD refinement, using the metrics of RMSE and MAE, respectively.

Method	Metrics	Lorenz	Plankton		Wind		Weather
			N4	N12	Osaka	Fukushima	
Arima+	RMSE (%)	0.45	0.05	0.28	0.413	0.18	1.41
	MAE (%)	0.35	0.02	0.00	0.09	0.22	1.15
ETS+	RMSE (%)	0.36	0.07	0.84	0.39	0.12	1.61
	MAE (%)	0.21	0.07	0.56	0.38	0.09	1.21
Theta+	RMSE (%)	0.42	0.09	1.25	0.49	0.43	3.16
	MAE (%)	0.31	0.09	1.00	0.051	0.59	2.16
SSA+	RMSE (%)	0.50	0.10	1.35	1.27	0.53	8.83
	MAE (%)	0.34	0.04	0.63	0.53	0.34	5.26
TRMF+	RMSE (%)	0.31	0.01	0.14	0.16	0.35	0.13
	MAE (%)	0.14	0.01	0.07	0.04	0.18	0.10
NBeats+	RMSE (%)	–	–	1.27	5.92	4.62	1.85
	MAE (%)	–	–	0.84	4.79	3.93	1.98
RNN+	RMSE (%)	–	0.14	0.70	0.87	–	2.22
	MAE (%)	–	0.02	0.92	0.79	–	1.62
Transformer+	RMSE (%)	–	0.08	0.75	0.51	0.62	0.39
	MAE (%)	–	0.00	0.35	0.58	0.07	0.21
MVE+	RMSE (%)	17.54	1.37	0.85	1.68	2.31	2.57
	MAE (%)	14.96	9.55	1.00	1.23	2.38	0.24
ARNN+	RMSE (%)	16.43	0.24	2.03	6.59	6.21	8.12
	MAE (%)	13.73	0.16	1.48	5.06	4.09	5.29
RDE+	RMSE (%)	0.58	0.04	0.50	1.29	0.05	2.03
	MAE (%)	0.55	0.03	0.35	1.40	0.07	1.40

Koopman operator to approximate the STI transformation function by estimating linear dynamical system states, providing a new formulation and tool for multi-step prediction of short-term time series. Further, we propose a unified optimization framework to solve the proposed STD model, which provides a unified understanding of previous models based on the STI framework. This novel model outperforms previous models in both simulated and real-world datasets. The results of refinement experiments show the possibility of refining previous prediction models with the proposed STD framework to improve their performance. The construction of Koopman invariant subspaces remains an open question for Koopman operator theory. Addressing this would further improve the STD performance.

Recently, deep learning methods and foundational models [59, 60] have made substantial strides, mainly in time-series prediction with long-term data. As a future study, we will consider combining the proposed STD framework with deep learning to construct multistep prediction models. Moreover, it is promising to develop methods by employing in-context operator learning [61], and simulating learning methodology based on the meta-learning paradigm [62, 63], to further enhance the STD model.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2022YFA1004100), in part by National Natural Science Foundation of China (Grant Nos. 62476214, 12326606, U24A20324, 32430017), in part by Tianyuan Fund for Mathematics of the National Natural Science Foundation of China (Grant No. 12426105), and in part by Major Key Project of Pengcheng Laboratory (Grant No. PCL2024A06).

Supporting information Appendixes A–G. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Luo Y, Ogle K, Tucker C, et al. Ecological forecasting and data assimilation in a data-rich era. Ecol Appl, 2011, 21: 1429–1442
2 Ye H, Beamish R J, Glaser S M, et al. Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. Proc Nat Academy Sci, 2015, 112: E1569–E1576
3 Stoffer D S, Ombao H. Editorial: special issue on time series analysis in the biological sciences. J Time Ser Anal, 2012, 33: 701–703
4 Buxton R B, Wong E C, Frank L R. Dynamics of blood flow and oxygenation changes during brain activation: the balloon model. Magn Reson Med, 1998, 39: 855–864

5 Mudelsee M. Trend analysis of climate time series: a review of methods. Earth-Sci Rev, 2019, 190: 310–322

6 He R, Zhang L, Chew A W Z. Data-driven multi-step prediction and analysis of monthly rainfall using explainable deep learning. Expert Syst Appl, 2024, 235: 121160

7 Li L, Qin L, Qu X, et al. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. Knowledge-Based Syst, 2019, 172: 1–14

8 Holt C C. Forecasting trends and seasonals by exponentially weighted moving averages. ONR Memorandum, 1957, 52: 5–10

9 Assimakopoulos V, Nikolopoulos K. The theta model: a decomposition approach to forecasting. Int J Forecasting, 2000, 16: 521–530

10 Box G E, Jenkins G M, Reinsel G C. Time Series Analysis. Hoboken: Wiley, 2008

11 Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: probabilistic forecasting with autoregressive recurrent networks. Int J Forecasting, 2020, 36: 1181–1191

12 Cenci S, Sugihara G, Saavedra S, et al. Regularized S-map for inference and forecasting with noisy ecological time series. Methods Ecol Evol, 2019, 10: 650–660

13 Liu Y, Hu T, Zhang H, et al. Itransformer: inverted transformers are effective for time series forecasting. In: Proceedings of the 12th International Conference on Learning Representations, 2024

14 Lin S, Lin W, Wu W, et al. Segrnn: segment recurrent neural network for long-term time series forecasting. 2023. ArXiv:2308.11200

15 Oreshkin B N, Carpov D, Chapados N, et al. N-beats: neural basis expansion analysis for interpretable time series forecasting. In: Proceedings of International Conference on Learning Representations, 2020

16 Munch S B, Rogers T L, Sugihara G. Recent developments in empirical dynamic modelling. Methods Ecol Evol, 2023, 14: 732–745

17 Sugihara G. Nonlinear forecasting for the classification of natural time series. Philos Trans A Math Phys Eng Sci, 1994, 348: 477–495

18 Ghadami A, Epureanu B I. Data-driven prediction in dynamical systems: recent developments. Phil Trans R Soc A, 2022, 380: 20210213

19 Venkatraman A, Hebert M, Bagnell J. Improving multi-step prediction of learned time series models. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2015

20 Ye H, Sugihara G. Information leverage in interconnected ecosystems: overcoming the curse of dimensionality. Science, 2016, 353: 922–925

21 Takens F. Detecting strange attractors in turbulence. In: Dynamical Systems and Turbulence. Berlin: Springer, 1981

22 Deyle E R, Sugihara G, Oresic M. Generalized theorems for nonlinear state space reconstruction. PLoS ONE, 2011, 6: e18295

23 Ma H, Zhou T, Aihara K, et al. Predicting time series from short-term high-dimensional data. Int J Bifurcation Chaos, 2014, 24: 1430033

24 Ma H, Leng S, Aihara K, et al. Randomly distributed embedding making short-term high-dimensional data predictable. Proc Nat Acad Sci, 2018, 115: E9994–E10002

25 Peng H, Chen P, Liu R, et al. Spatiotemporal information conversion machine for time-series forecasting. Fundamental Res, 2024, 4: 1674–1687

26 Peng H, Wang W, Chen P, et al. DEFM: delay-embedding-based forecast machine for time series forecasting by spatiotemporal information transformation. Chaos-An Interdisciplinary J Nonlinear Sci, 2024, 34: 043112

27 Peng H, Chen P, Yang N, et al. One-core neuron deep learning for time series prediction. Natl Sci Rev, 2025, 12: nwae441

28 Chen P, Liu R, Aihara K, et al. Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation. Nat Commun, 2020, 11: 4568

29 Breiman L. Bagging predictors. Mach Learn, 1996, 24: 123–140

30 Tanaka G, Yamane T, Héroux J B, et al. Recent advances in physical reservoir computing: a review. Neural Netws, 2019, 115: 100–123

31 Tong Y, Hong R, Zhang Z, et al. Earthquake alerting based on spatial geodetic data by spatiotemporal information transformation learning. Proc Natl Acad Sci USA, 2023, 120: e2302275120

32 Koopman B O. Hamiltonian systems and transformation in Hilbert space. Proc Natl Acad Sci USA, 1931, 17: 315–318

33 Koopman B O, Neumann J. Dynamical systems of continuous spectra. Proc Natl Acad Sci USA, 1932, 18: 255–263

34 Netto M, Mili L. A robust data-driven Koopman Kalman filter for power systems dynamic state estimation. IEEE Trans Power Syst, 2018, 33: 7228–7237

35 Mezić I. Spectrum of the Koopman operator, spectral expansions in functional spaces, and state-space geometry. J Nonlinear Sci, 2020, 30: 2091–2145

36 Lee K, Carlberg K T. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J Comput Phys, 2020, 404: 108973

37 Redman W T. On Koopman mode decomposition and tensor component analysis. Chaos-An Interdisciplinary J Nonlinear Sci, 2021, 31: 051101

38 Otto S E, Rowley C W. Koopman operators for estimation and control of dynamical systems. Annu Rev Control Robot Auton Syst, 2021, 4: 59–87

39 Rosenfeld J A, Kamalapurkar R, Gruss L F, et al. Dynamic mode decomposition for continuous time systems with the liouville operator. J Nonlinear Sci, 2022, 32: 1–30

40 Brunton S L, Budišić M, Kaiser E, et al. Modern Koopman theory for dynamical systems. SIAM Rev, 2022, 64: 229–340

41 Rowley C W, Mezić I, Bagheri S, et al. Spectral analysis of nonlinear flows. J Fluid Mech, 2009, 641: 115–127

42 Williams M O, Kevrekidis I G, Rowley C W. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. J Nonlinear Sci, 2015, 25: 1307–1346

43 Azriel D, Brown L D, Sklar M, et al. Semi-supervised linear regression. J Am Stat Assoc, 2022, 117: 2238–2251

44 Tibshirani R. Regression shrinkage and selection via the Lasso. J R Stat Soc Ser B-Stat Methodol, 1996, 58: 267–288

45 Golub G H, Pereyra V. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM J Numer Anal, 1973, 10: 413–432

46 Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag Sci, 2009, 2: 183–202

47 Gillard J, Usevich K. Hankel low-rank approximation and completion in time series analysis and forecasting: a brief review. Stat Its Interface, 2023, 16: 287–303

48 Baddoo P J, Herrmann B, McKeon B J, et al. Kernel learning for robust dynamic mode decomposition: linear and nonlinear disambiguation optimization. Proc R Soc A, 2022, 478: 20210830

49 Lusch B, Kutz J N, Brunton S L. Deep learning for universal linear embeddings of nonlinear dynamics. Nat Commun, 2018, 9: 4950

50 Rahimi A, Recht B. Random features for large-scale kernel machines. In: Proceedings of Advances in Neural Information Processing Systems, 2007

51 Paszke A, Gross S, Massa F, et al. Pytorch: an imperative style, high-performance deep learning library. In: Proceedings of Advances in Neural Information Processing Systems, 2019

52 Federico G, Max Mergenthaler C, Cristian C, et al. StatsForecast: lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, US 2022, 2022

53 Agarwal A, Amjad M J, Shah D, et al. Model agnostic time series analysis via matrix estimation. Proc ACM Meas Anal Comput Syst, 2018, 2: 1–39

54 Yu H F, Rao N, Dhillon I S. Temporal regularized matrix factorization for high-dimensional time series prediction. In: Proceedings of Advances in Neural Information Processing Systems, 2016, 29

55 Benincà E, Huisman J, Heerkloss R, et al. Chaos in a long-term experiment with a plankton community. Nature, 2008, 451: 822–825

56 Hirata Y, Aihara K. Predicting ramps by integrating different sorts of information. Eur Phys J Spec Top, 2016, 225: 513–525

57 Clark A T, Ye H, Isbell F, et al. Spatial convergent cross mapping to detect causal relationships from short time series. Ecology, 2015, 96: 1174–1181

58 Makridakis S, Spiliotis E, Assimakopoulos V, et al. Statistical and machine learning forecasting methods: concerns and ways forward. PLoS ONE, 2018, 13: e0194889

59 Jin M, Wen Q, Liang Y, et al. Large models for time series and spatio-temporal data: a survey and outlook. 2023. ArXiv:2310.10196

60 Gruver N, Finzi M, Qiu S, et al. Large language models are zero-shot time series forecasters. In: Proceedings of Advances in Neural Information Processing Systems, 2024, 36

61 Yang L, Liu S, Meng T, et al. In-context operator learning with data prompts for differential equation problems. Proc Natl Acad Sci USA, 2023, 120: e2310142120

62 Shu J, Meng D, Xu Z. Learning an explicit hyperparameter prediction function conditioned on tasks. J Mach Learn Res, 2023, 24: 8818–8891

63 Xu Z, Shu J, Meng D. Simulating learning methodology (SLeM): an approach to machine learning automation. Natl Sci Rev, 2024, 11: nwae277