



## OPEN ACCESS

## EDITED BY

Roberto Interdonato,  
UMR9000 Territoires, Environnement,  
Téledétection et Information Spatiale (TETIS),  
France

## REVIEWED BY

Fei Yu,  
Changsha University of Science and  
Technology, China  
Matteo Sangiorgio,  
Polytechnic University of Milan, Italy

## \*CORRESPONDENCE

Jiming Liu  
✉ [jiming@comp.hkbu.edu.hk](mailto:jiming@comp.hkbu.edu.hk)

RECEIVED 05 October 2024

ACCEPTED 09 December 2024

PUBLISHED 17 January 2025

## CITATION

Feng L, Liu Y, Shi B and Liu J (2025) Toward a  
physics-guided machine learning approach  
for predicting chaotic systems dynamics.  
*Front. Big Data* 7:1506443.  
doi: 10.3389/fdata.2024.1506443

## COPYRIGHT

© 2025 Feng, Liu, Shi and Liu. This is an  
open-access article distributed under the  
terms of the [Creative Commons Attribution  
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or  
reproduction in other forums is permitted,  
provided the original author(s) and the  
copyright owner(s) are credited and that the  
original publication in this journal is cited, in  
accordance with accepted academic practice.  
No use, distribution or reproduction is  
permitted which does not comply with these  
terms.

# Toward a physics-guided machine learning approach for predicting chaotic systems dynamics

Liu Feng<sup>1</sup>, Yang Liu<sup>1</sup>, Benyun Shi<sup>2</sup> and Jiming Liu<sup>1\*</sup>

<sup>1</sup>Department of Computer Science, Hong Kong Baptist University, Hong Kong, China, <sup>2</sup>College of Computer and Information Engineering, Nanjing Tech University, Nanjing, China

Predicting the dynamics of chaotic systems is crucial across various practical domains, including the control of infectious diseases and responses to extreme weather events. Such predictions provide quantitative insights into the future behaviors of these complex systems, thereby guiding the decision-making and planning within the respective fields. Recently, data-driven approaches, renowned for their capacity to learn from empirical data, have been widely used to predict chaotic system dynamics. However, these methods rely solely on historical observations while ignoring the underlying mechanisms that govern the systems' behaviors. Consequently, they may perform well in short-term predictions by effectively fitting the data, but their ability to make accurate long-term predictions is limited. A critical challenge in modeling chaotic systems lies in their sensitivity to initial conditions; even a slight variation can lead to significant divergence in actual and predicted trajectories over a finite number of time steps. In this paper, we propose a novel Physics-Guided Learning (PGL) method, aiming at extending the scope of accurate forecasting as much as possible. The proposed method aims to synergize observational data with the governing physical laws of chaotic systems to predict the systems' future dynamics. Specifically, our method consists of three key elements: a data-driven component (DDC) that captures dynamic patterns and mapping functions from historical data; a physics-guided component (PGC) that leverages the governing principles of the system to inform and constrain the learning process; and a nonlinear learning component (NLC) that effectively synthesizes the outputs of both the data-driven and physics-guided components. Empirical validation on six dynamical systems, each exhibiting unique chaotic behaviors, demonstrates that PGL achieves lower prediction errors than existing benchmark predictive models. The results highlight the efficacy of our design of data-physics integration in improving the precision of chaotic system dynamics forecasts.

## KEYWORDS

physics-guided, data-driven, deep learning, chaotic systems, dynamics prediction

## 1 Introduction

Chaotic systems are ubiquitous, from academic research in physics (Pecora and Carroll, 1990; Grassberger and Procaccia, 1983) and chemistry (Hess, 1990; Field et al., 1993) to real-world domains such as epidemiology (Aguar et al., 2008; Mishra et al., 2020) and climatology (Palmer, 1993; Olsen et al., 2019). By predicting the dynamics of these systems, we can gain valuable insights into their future behaviors, which can not only

help us understand the underlying mechanisms of these systems but, more importantly, effectively inform and guide the decision-making process in real-world problems within the respective fields. For example, forecasting the dynamical behaviors in the spread of epidemics can help us uncover the disease transmission patterns and, accordingly, deploy effective intervention strategies to control infectious diseases (Mangiarotti et al., 2016). Predicting the dynamics of variables in the climate system, such as temperature and precipitation, can help us be well prepared for extreme weather events (Toreti et al., 2013).

In recent years, with the availability of large amounts of data and the advancement of computing power, many studies have utilized data-driven approaches to analyze and predict the dynamics of chaotic systems. These methods generally utilize the given data to learn the mapping function between historical observations and the future value of the target variable, and then use the learned mapping function to conduct the prediction. Typical data-driven methods that have been widely used in chaotic system dynamics prediction include long short-term memory networks (LSTM) (Hochreiter, 1997; Chattopadhyay et al., 2020), reservoir computing (Jaeger, 2001; Pathak et al., 2018), etc. The above methods have been proven to be effective for the short-term prediction of chaotic systems, demonstrating an ability to capture the instantaneous dynamics (Chantry et al., 2021). However, their ability to make long-term predictions is limited, especially for those rapidly evolving chaotic dynamical systems, where even a slight initial variation can result in significant differences as the evolution over time (Lorenz, 1963). The reason could be that such data-driven methods rely solely on historical observations during the learning process but ignore the underlying mechanisms of chaotic systems, which are, in fact, of great importance in characterizing the systems' dynamical behaviors.

To overcome the limitations of pure data-driven models in predicting chaotic system dynamics and to enhance prediction performance, several existing studies have combined data with physical mechanisms. For example, PIESN (Doan et al., 2020) and its variant (Na et al., 2023) encode the systems' governing equations into the models' loss functions, penalizing predictions that deviate from physical laws. Furthermore, other methods utilize physical knowledge to help reconstruct and predict the dynamics of chaotic systems with unmeasured variables (Racca and Magri, 2021; Özalp et al., 2023). These methods, however, typically require complete and precise knowledge of the governing differential equations of the systems, including the equation parameters, to effectively guide the predictive models, which limits their applicability. Meanwhile, the reconciliation between data-driven approaches and prior physical knowledge remains an open yet essential problem in the prediction of chaotic systems' dynamics.

To effectively extend the capability for chaotic dynamics prediction, in this paper, we introduce a novel method called Physics-Guided Learning (PGL). Inspired by a recently developed physics-informed neural network (PINN), which was originally designed for solving forward and reverse problems in nonlinear partial differential equations (Raissi et al., 2019), our PGL method seeks to synergize observational data with the governing physical laws of chaotic systems. In our study, we operate under the

assumption that the knowledge of the dynamical system we aim to predict is partially available. Specifically, we assume familiarity with the structure of the ordinal differential equations, while the parameters of these equations remain unknown and will be inferred throughout the learning process. This modest assumption has been widely adopted in recent research in physics-informed machine learning and aligns with many real-world scenarios where precise governing equations are not accessible (Misyris et al., 2020; Nath et al., 2023; Ning et al., 2023). For example, in climate modeling, researchers often rely on the well-established Navier-Stokes equations, despite the challenges in determining their exact parameters and solutions (Yang et al., 2023; Gao et al., 2024). The architecture of PGL is composed of three integral components: a data-driven component that learns the dynamical patterns and mapping functions from historical observations, a physics-guided component that exploits and represents systems' governing mechanisms, and a nonlinear learning component that integrates the output from the data-driven component and that from the physics-guided component in a proper way. The objective functions of these three components will be jointly optimized to achieve the desired goal of chaotic dynamics prediction.

Several related works have explored the use of neural networks to generate chaotic dynamics. Notably, Hopfield Neural Networks (Hopfield, 1984) with memristors (Chua, 1971) have attracted much attention due to their flexible network architecture and bio-inspired characteristics. These models have been employed to produce a variety of chaotic dynamics, including multi-scroll, coexisting, and hyperchaotic attractors (Li et al., 2022; Kong et al., 2024; Deng et al., 2024). In contrast to approaches that generate dynamics with chaotic characteristics for applications such as image encryption (Liu et al., 2019) and privacy protection (Hu et al., 2024), and that do not necessitate reference to a specific dynamical system, our study seeks to predict the dynamical behaviors of a particular chaotic system. We employ data-driven methods, specifically neural networks, leveraging historical observations and partial knowledge of the chaotic system being modeled. By integrating data with physical principles, we aim to extend the scope and accuracy of chaotic dynamics prediction.

The remainder of this paper is organized as follows. Section 2 outlines the proposed methodology, with a detailed explanation of its core principles, architecture design, and learning processes. In Section 3, we present the settings and results of our experiments on six typical chaotic systems, which are designed to validate the effectiveness of the proposed method in the task of chaotic dynamics prediction. Finally, we conclude our work in Section 4.

## 2 Methodology

In this section, we will outline the formalism and computational mechanism of the proposed PGL method. We begin by defining the learning problem and providing an overview of the method. Subsequently, we present the mathematical definition and formulation of the proposed method for chaotic system dynamics prediction, which integrates data and physical understanding. To enhance the clarity, we detail the method's structure, workflow, and objective function.

## 2.1 Problem statement

First, we state the definition of chaotic system dynamics prediction. For a chaotic system with  $N$  state variables, we represent the system's state observations at time  $t$  as  $\mathbf{X}_t = [x_t^1, x_t^2, \dots, x_t^N]$ .  $\mathbf{X}_{t-L+1:t} = [\mathbf{X}_{t-L+1}, \mathbf{X}_{t-L+2}, \dots, \mathbf{X}_t]$  denotes the historical data containing  $L$  time steps. Meanwhile, the time point sequence  $\mathbf{T}_{t-L+1:t} = [t-L+1, t-L+2, \dots, t]$  corresponding to the system's state value sequence  $\mathbf{X}_{t-L+1:t}$  is also recorded. The target of chaotic system dynamics prediction is to learn the underlying state transition function and the potential dynamics of the system based on the historical data and governing physical laws, and then forecast the subsequent state of the chaotic system, denoted as  $\tilde{\mathbf{X}}_{t+1}$ . To achieve this goal, we devise a PGL method that makes use of both the observational data and the underlying dynamical mechanism of the chaotic system. Specifically, the proposed method comprises three core components: a data-driven component (DDC), a physics-guided component (PGC), and a nonlinear learning component (NLC). In the subsequent section, we will furnish a more detailed exposition of our design.

## 2.2 Physics-guided learning

Figure 1 illustrates the architecture of the proposed method PGL, consisting of DDC, PGC, and NLC. For the DDC, we use a three-layer LSTM with 20 hidden units each, followed by a dense layer. For the PGC, we refer to the PINN configuration (Raissi et al., 2019), using a 10-layer neural network with 32 neurons in each layer. For the third component NLC, note that it is intentionally designed to affirm the feasibility of the proposed idea of integrating data-driven and physics-guided components. Due to the real-world data often exhibits different complex nonlinear patterns, our model, which can be seen as a physics-guided learning framework, is designed with flexibility, allowing for the incorporation of different sophisticated neural network architectures to accommodate and adapt to these higher levels of complexity. In this paper, we utilize two typical architectures—the multi-layer perceptron (MLP)<sup>1</sup> and the attention mechanism—as examples to demonstrate our design of the NLC. Specifically, the MLP-based NLC has two layers: one input layer and one output layer. In the attention-based NLC, we utilize the cross-attention mechanism to capture the nonlinearity in the DDC and PGC's outputs (Vaswani, 2017; Shi et al., 2024). Note that other deep learning modules or architectures can also be flexibly integrated into our framework as the NLC. Next, we will elaborate in detail on how these three components work together to predict the dynamical behaviors of chaotic systems.

### 2.2.1 Data-driven component

Firstly, we obtain the prediction of the data-driven branch for the next time step, denoted by  $\mathbf{X}_{t+1}^{data} = DDC(\mathbf{X}_{t-L+1:t})$ . We expect the long short-term memory (LSTM) structure in the DDC to

<sup>1</sup> A preliminary version of this work appeared in the 4th French Regional Conference on Complex Systems (FRCCS 2024) (Feng et al., 2024).

capture both short-term and long-term temporal dependencies in the historical state sequence through its unique gating mechanism and make predictions for the next time step.

### 2.2.2 Physics-guided component

Afterward, we extend the  $\mathbf{T}_{t-L+1:t}$ , turning it into  $\mathbf{T}_{t-L+1:t+L}$ , which is further fed into the PGC. The PGC generates the system state predictions that are of equal length to the extended time sequence  $\mathbf{T}_{t-L+1:t+L}$ . This process is shown in the following equation:

$$\mathbf{X}_{t-L+1:t+L}^{phy} = PGC(t-L+1, t-L+2, \dots, t+L), \quad (1)$$

where  $\mathbf{X}_i^{phy} = [x_i^{phy}, y_i^{phy}, z_i^{phy}]$ . We expect that, with the guidance of physical knowledge, the PGC can learn the dynamics of the system and assist the entire model in making predictions. Note that the design of PGC is general and can be used in various chaotic systems. Here, for a better explanation, we use the typical Lorenz system (Lorenz, 1963) as an example to show how the PGC works. The only information that we have is the form of the system's equations shown in the following Equation 2, and we do not know the crucial initial values and system parameters.

$$\begin{aligned} \frac{dx}{dt} &= a(y-x), \\ \frac{dy}{dt} &= cx-y-xz, \\ \frac{dz}{dt} &= xy-bz. \end{aligned} \quad (2)$$

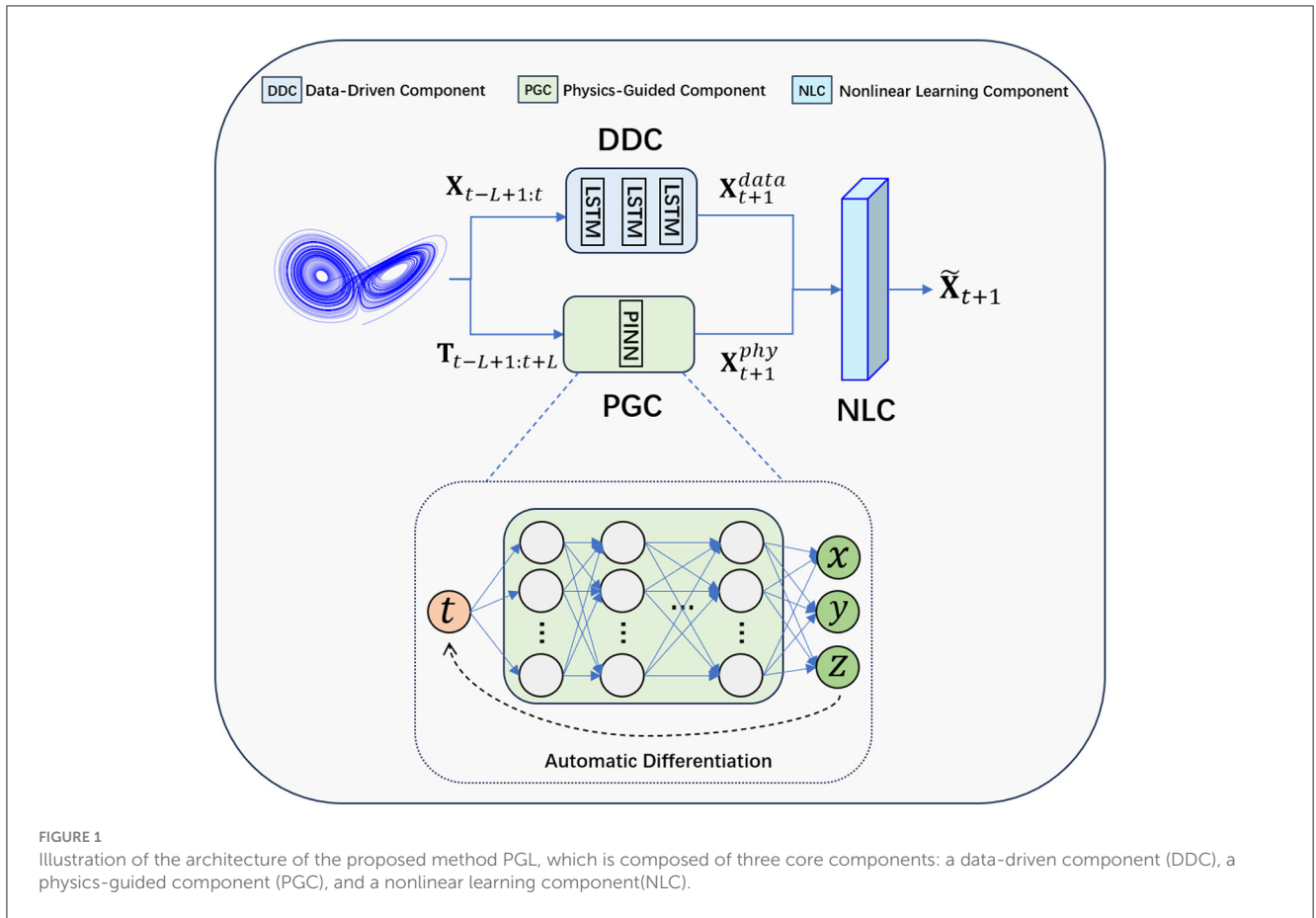
Following the work of physics-informed neural networks in Raissi et al. (2019), we utilize the automatic differentiation tools within the deep learning framework PyTorch (Paszke et al., 2017) to compute the derivative of the PGC's output  $\mathbf{X}_{t-L+1:t+L}^{phy}$  with respect to its input  $\mathbf{T}_{t-L+1:t+L}$ , yielding the following:

$$\frac{\partial \mathbf{X}_{t-L+1:t+L}^{phy}}{\partial \mathbf{T}} = \left[ \frac{\partial \mathbf{X}_{t-L+1}^{phy}}{\partial \mathbf{T}}, \frac{\partial \mathbf{X}_{t-L+2}^{phy}}{\partial \mathbf{T}}, \dots, \frac{\partial \mathbf{X}_{t+L}^{phy}}{\partial \mathbf{T}} \right], \quad (3)$$

where  $\frac{\partial \mathbf{X}_i^{phy}}{\partial \mathbf{T}} = \left[ \frac{\partial x_i^{phy}}{\partial \mathbf{T}}, \frac{\partial y_i^{phy}}{\partial \mathbf{T}}, \frac{\partial z_i^{phy}}{\partial \mathbf{T}} \right]$ . We expect that the approximate derivatives conform to the definition of the Lorenz system, and therefore, we have calculated the residuals with respect to the physics-guided component, as shown below.

$$\begin{aligned} loss_{phy} &= \lambda_1 loss_x + \lambda_2 loss_y + \lambda_3 loss_z, \\ loss_x &= \sum_{i=t-L+1}^{t+L} \left| \frac{\partial x_i^{phy}}{\partial \mathbf{T}} - \tilde{a} (y_i^{phy} - x_i^{phy}) \right|^2, \\ loss_y &= \sum_{i=t-L+1}^{t+L} \left| \frac{\partial y_i^{phy}}{\partial \mathbf{T}} - (\tilde{c} x_i^{phy} - y_i^{phy} - x_i^{phy} z_i^{phy}) \right|^2, \\ loss_z &= \sum_{i=t-L+1}^{t+L} \left| \frac{\partial z_i^{phy}}{\partial \mathbf{T}} - (x_i^{phy} y_i^{phy} - \tilde{b} z_i^{phy}) \right|^2, \end{aligned} \quad (4)$$

where  $\lambda_1, \lambda_2$ , and  $\lambda_3$  are hyper parameters which can be selected by a grid search strategy from a predefined rough range in practice.  $\tilde{a}$ ,



$\tilde{b}$ , and  $\tilde{c}$  are trainable parameters of the model. Note that the true parameters of the chaotic systems remain unidentified for the PGC and for the proposed PGL model, a scenario that is typical in real-world applications. It is our expectation that the proposed model is capable of learning and characterizing the systems' dynamics even in the presence of such uncertainties. Additionally, since we have the ground truth  $\mathbf{X}_{t-L+1:t}$ , we conduct supervised learning by minimizing the following  $loss_{data}$ :

$$loss_{data} = \frac{1}{L} \sum_{i=t-L+1}^t \left| \mathbf{X}_i^{phy} - \mathbf{X}_i \right|^2. \tag{5}$$

By incorporating penalty terms based on physics and data, we hope that the PGC can rely on known physical knowledge and work in collaboration with the DDC to predict chaotic systems.

### 2.2.3 Nonlinear learning component

Next, a nonlinear learning component will balance the predicted  $\mathbf{X}_{t+1}^{data}$  and  $\mathbf{X}_{t+1}^{phy}$  from DDC and PGC to provide the final prediction  $\tilde{\mathbf{X}}_{t+1}$  for the system at the time step  $t + 1$ . In the following, we will introduce the MLP-based NLC and the Attention-based NLC, separately.

#### 2.2.3.1 MLP-based NLC

In the MLP-based NLC, we utilize a classical structure of MLP to conduct the nonlinear learning task, which can be described as

the following equation:

$$\tilde{\mathbf{X}}_{t+1} = NLC \left( concatenate \left( \mathbf{X}_{t+1}^{data}, \mathbf{X}_{t+1}^{phy} \right) \right), \tag{6}$$

where  $\tilde{\mathbf{X}}_{t+1}$  represents the predicted value for the next time step. To constraints the learning process, we also calculate the loss which is formulated as follows:

$$loss_{NLC} = \left| \tilde{\mathbf{X}}_{t+1} - \mathbf{X}_{t+1} \right|^2, \tag{7}$$

where  $\mathbf{X}_{t+1}$  denotes the ground truth value of the system's state variable at time step  $t + 1$ , which serves as the label in our supervised learning. It is important to note that the data for  $\mathbf{X}_{t+1}$  in Equation 7 is exclusively accessible during the training phase. This information is not available during the testing phase, where the model must predict  $\mathbf{X}_{t+1}$  without the aid of ground truth values.

#### 2.2.3.2 Attention-based NLC

In the Attention-based NLC, we use a specifically designed attention mechanism, i.e. the cross-attention, to learn the nonlinearity and make the final predictions. First, the attention mechanism generates the query  $Q_{data}$ , the key  $K_{data}$ , and the value  $V_{data}$  by applying linear transformations to  $\mathbf{X}_{t+1}^{data}$ , i.e.,  $Q_{data} = \mathbf{X}_{t+1}^{data} \cdot W_q^{data}$ ,  $K_{data} = \mathbf{X}_{t+1}^{data} \cdot W_k^{data}$ , and  $V_{data} = \mathbf{X}_{t+1}^{data} \cdot W_v^{data}$ , where  $W_q^{data}$ ,  $W_k^{data}$ , and  $W_v^{data}$  are the trainable matrices. Similarly, we can obtain the query  $Q_{phy}$ , the key  $K_{phy}$ , and the value  $V_{phy}$  by performing the same calculation for the output of PGC  $\mathbf{X}_{t+1}^{phy}$ . Then,

we can further calculate the attention feature maps  $A^{data}$  and  $A^{phy}$  based on these units:

$$\begin{aligned} A^{data} &= \text{softmax}\left(\frac{Q_{data}K_{data}^T}{\sqrt{d_{K_{data}}}}\right) \cdot V_{data} + \mathbf{X}_{t+1}^{data}, \\ A^{phy} &= \text{softmax}\left(\frac{Q_{phy}K_{phy}^T}{\sqrt{d_{K_{phy}}}}\right) \cdot V_{phy} + \mathbf{X}_{t+1}^{phy}, \end{aligned} \quad (8)$$

where  $d_{K_{data}}$  and  $d_{K_{phy}}$  denote the dimensions of  $K_{data}$  and  $K_{phy}$ , respectively, and softmax is an activation function. By doing so, we intend to learn the important information in the outputs from DDC and PGC separately, so as to guarantee the prediction performance.

Next, we attempt to capture the nonlinear relationships between  $\mathbf{X}_{t+1}^{data}$  and  $\mathbf{X}_{t+1}^{phy}$  by applying the cross-attention mechanism. The cross-attention feature map  $CA^{data}$  can be obtained by using  $Q_{phy}$  to query the key-value pair  $(K_{data}, V_{data})$ :

$$CA^{data} = \text{softmax}\left(\frac{Q_{phy}K_{data}^T}{\sqrt{d_{K_{data}}}}\right) \cdot V_{data} + \mathbf{X}_{t+1}^{data}. \quad (9)$$

Similarly, the cross-attention feature map  $CA^{phy}$  can be calculated as follows:

$$CA^{phy} = \text{softmax}\left(\frac{Q_{data}K_{phy}^T}{\sqrt{d_{K_{phy}}}}\right) \cdot V_{phy} + \mathbf{X}_{t+1}^{phy}. \quad (10)$$

Finally, all the feature maps obtained above are concatenated and fed into the output layer, to make the final prediction  $\tilde{\mathbf{X}}_{t+1}$ :

$$\tilde{\mathbf{X}}_{t+1} = F\left(\text{concatenate}\left(A^{data}, A^{phy}, CA^{data}, CA^{phy}\right)\right), \quad (11)$$

where  $F$  denotes the output layer in the Attention-based NLC. Same as MLP-based NLC, we also calculate the loss  $loss_{NLC} = |\tilde{\mathbf{X}}_{t+1} - \mathbf{X}_{t+1}|^2$ , to constraint the learning process.

### 2.2.4 Objective function

The final optimization objective function, which takes account of both data and physics, is given as follows:

$$\min(w_1 loss_{NLC} + w_2 loss_{data} + w_3 loss_{phy}), \quad (12)$$

where  $w_1$ ,  $w_2$ , and  $w_3$  are hyper parameters.

## 3 Experimental results

In this section, we use six dynamical systems with different chaotic behaviors, i.e., the Rossler, Aizawa, Lorenz, Chua, Chen, and Halvorsen systems, which are widely used in chaotic systems dynamics prediction (Nasiri and Ebadzadeh, 2022; Cheng et al., 2021; Na et al., 2021; Wu et al., 2024; Kennedy et al., 2024; Gilpin, 2021), to validate the performance of the proposed PGL method in long-term forecasting of chaotic dynamics. We also perform an ablation study to analyze the contributions of different components of the proposed method to the chaotic dynamics prediction.

## 3.1 Descriptions of chaotic systems

### 3.1.1 Rossler system

In 1976, Rössler (1976) proposed the well-known Rossler system, which exhibits chaotic phenomena and nonlinear dynamical behavior. The system is defined by the following differential equations:

$$\begin{aligned} \frac{dx}{dt} &= -y - z, \\ \frac{dy}{dt} &= x - ay, \\ \frac{dz}{dt} &= b + xz - cz. \end{aligned} \quad (13)$$

### 3.1.2 Aizawa system

In 1982, Aizawa and Uezu (1982) introduced a new chaotic system, which has multiple three-order nonlinear terms. The Aizawa system can be described by the following equations:

$$\begin{aligned} \frac{dx}{dt} &= (z - b)x - dy, \\ \frac{dy}{dt} &= dx + (z - b)y, \\ \frac{dz}{dt} &= c + az - \frac{z^3}{3} - (x^2 + y^2)(1 + ez) + fzx^3. \end{aligned} \quad (14)$$

### 3.1.3 Lorenz system

In 1963, Lorenz (1963) discovered the existence of a peculiar “butterfly effect” in meteorological systems when studying convective instability. The Lorenz system can be described by the following equations:

$$\begin{aligned} \frac{dx}{dt} &= a(y - x), \\ \frac{dy}{dt} &= cx - y - xz, \\ \frac{dz}{dt} &= xy - bz. \end{aligned} \quad (15)$$

### 3.1.4 Chua system

In 1986, Chua et al. (1986) introduced the Chua system, marking an advancement in the study of chaotic systems by linking chaos and nonlinear circuits. The equations of the Chua system are given as follows:

$$\begin{aligned} \frac{dx}{dt} &= a(y - x - G(x)), \\ \frac{dy}{dt} &= x - y + z, \\ \frac{dz}{dt} &= -by, \\ G(x) &= cx + (d + c)(|x + 1| - |x - 1|). \end{aligned} \quad (16)$$

### 3.1.5 Chen system

In 1999, Chen and Ueta (1999) identified a chaotic attractor that bears similarities to the Lorenz system but is topologically

TABLE 1 The system parameters and initial values of six chaotic systems used in our study.

System	Parameters	Initial values
Rosler	$a = 0.2, b = 0.2, c = 5.7$	$(x_0, y_0, z_0) = (1.0, 1.0, 1.0)$
Aizawa	$a = 0.95, b = 0.7, c = 0.6, d = 3.5, e = 0.25, f = 0.1$	$(x_0, y_0, z_0) = (0.1, 0, 0.1)$
Lorenz	$a = 10.0, b = 8/3, c = 28.0$	$(x_0, y_0, z_0) = (1.0, 1.0, 1.0)$
Chua	$a = 15.6, b = 25.28, c = -0.75, d = 0.47$	$(x_0, y_0, z_0) = (0.1, 0.1, 0.1)$
Chen	$a = 35.0, b = 3.0, c = 28.0$	$(x_0, y_0, z_0) = (0, 1.0, 0)$
Halvorsen	$a = 1.4$	$(x_0, y_0, z_0) = (1.0, 0, 0)$

distinct in their research on chaotic control. The Chen system can be described by the following equations:

$$\begin{aligned} \frac{dx}{dt} &= a(y - x), \\ \frac{dy}{dt} &= (c - a)x - xz + cy, \\ \frac{dz}{dt} &= xy - bz. \end{aligned} \quad (17)$$

### 3.1.6 Halvorsen System

The [Sprott \(2010\)](#) system, proposed by Arne Dehli Halvorsen, is a 3-D system of chaotic flows whose governing equations are cyclically symmetric and can be described as follows:

$$\begin{aligned} \frac{dx}{dt} &= -ax - 4y - 4z - y^2, \\ \frac{dy}{dt} &= -ay - 4z - 4x - z^2, \\ \frac{dz}{dt} &= -az - 4x - 4y - x^2. \end{aligned} \quad (18)$$

All the above six dynamical systems have nonlinear and chaotic behaviors, posing great challenges for long-term prediction. We use the fourth-order Runge-Kutta method with a step size of 0.01 to obtain the chaotic time series containing 10,000 steps, which are divided into training, validation, and testing datasets in a ratio of 6:2:2. Specifically, we utilize the data from the initial 6,000 time steps for training purposes. This is followed by the subsequent 2,000 time steps, which are designated for the validation process. Finally, we employ the data from the concluding 2,000 time steps to test the performance of our model. [Table 1](#) provides the details of system parameters and initial values. For parameters  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  in [Equation 4](#) of the proposed method, we determine their values through a grid search strategy. Specifically, the parameter values are empirically constrained within the range of [0.05, 0.35], with a search step size of 0.05.

## 3.2 Comparison models and evaluation metrics

We select five representative methods as the baselines for performance comparison in our experiments. They are the

long short-term memory (LSTM) ([Hochreiter, 1997](#)), the echo state network (ESN) ([Pathak et al., 2017](#)), the next generation reservoir computing method (NG-RC) ([Gauthier et al., 2021](#)), the knowledge-based neural ordinary differential equations method (K-NODE) ([Jiahao et al., 2021](#)) and DLinear ([Zeng et al., 2023](#)). Here, LSTM is a classic recurrent neural network model for time series prediction; ESN and NG-RC are representative methods specifically designed and widely used for chaotic system dynamics prediction; DLinear is a state-of-the-art deep learning method developed for complex time series forecasting; and K-NODE is a hybrid-learning approach which integrates the first principles knowledge, specifically the ordinary differential equations, with data-driven technologies, to predict chaotic systems dynamics. For LSTM, we use a three-layer architecture with a uniform hidden state size. To achieve its optimal performance, we experiment with a variety of hidden state sizes, specifically 8, 16, and 32, and report the best result. For ESN, we implement it with a spectral radius of 1.4 and a reservoir size of 300. For NG-RC and DLinear, we follow the default settings reported in their original papers. For K-NODE, we set the prior knowledge as the form of the governing equations with the approximated parameters learned by classic symbolic regression.

When assessing the effectiveness of the methods in capturing and forecasting the dynamical behavior of chaotic systems over the long term, it is a common practice to employ the model's own prediction as the input for forecasting subsequent time steps during the test phase. This iterative process can result in an increase in errors as the forecast horizon extends, especially in chaotic systems, where small deviations at the beginning can lead to significant differences in later outcomes. The mean absolute error (MAE), the root mean square error (RMSE), and the  $R^2$  ([Amaranto and Mazzoleni, 2023](#)) are used as evaluation metrics to measure the prediction performance. The MAE and RMSE are defined as follows:

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T |\hat{y}_t - y_t|, \quad (19)$$

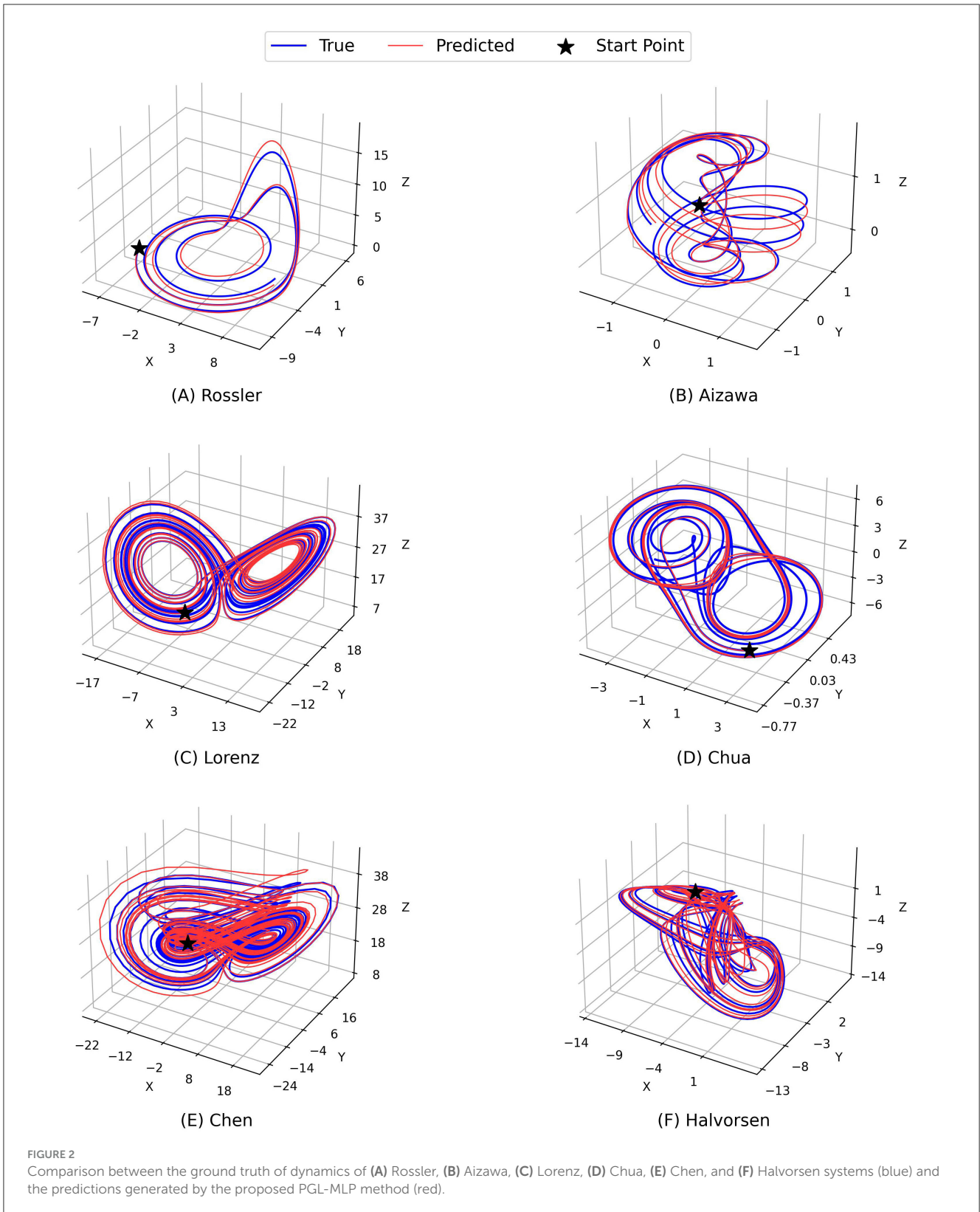
$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T (\hat{y}_t - y_t)^2}, \quad (20)$$

$$R^2 = 1 - \frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{\sum_{t=1}^T (\bar{y} - y_t)^2} \quad (21)$$

where  $\hat{y}_t$  denotes the predicted value of the model,  $y_t$  denotes the ground truth,  $\bar{y}$  represents the average value of the ground truth, and  $T$  is the corresponding forecast horizon.

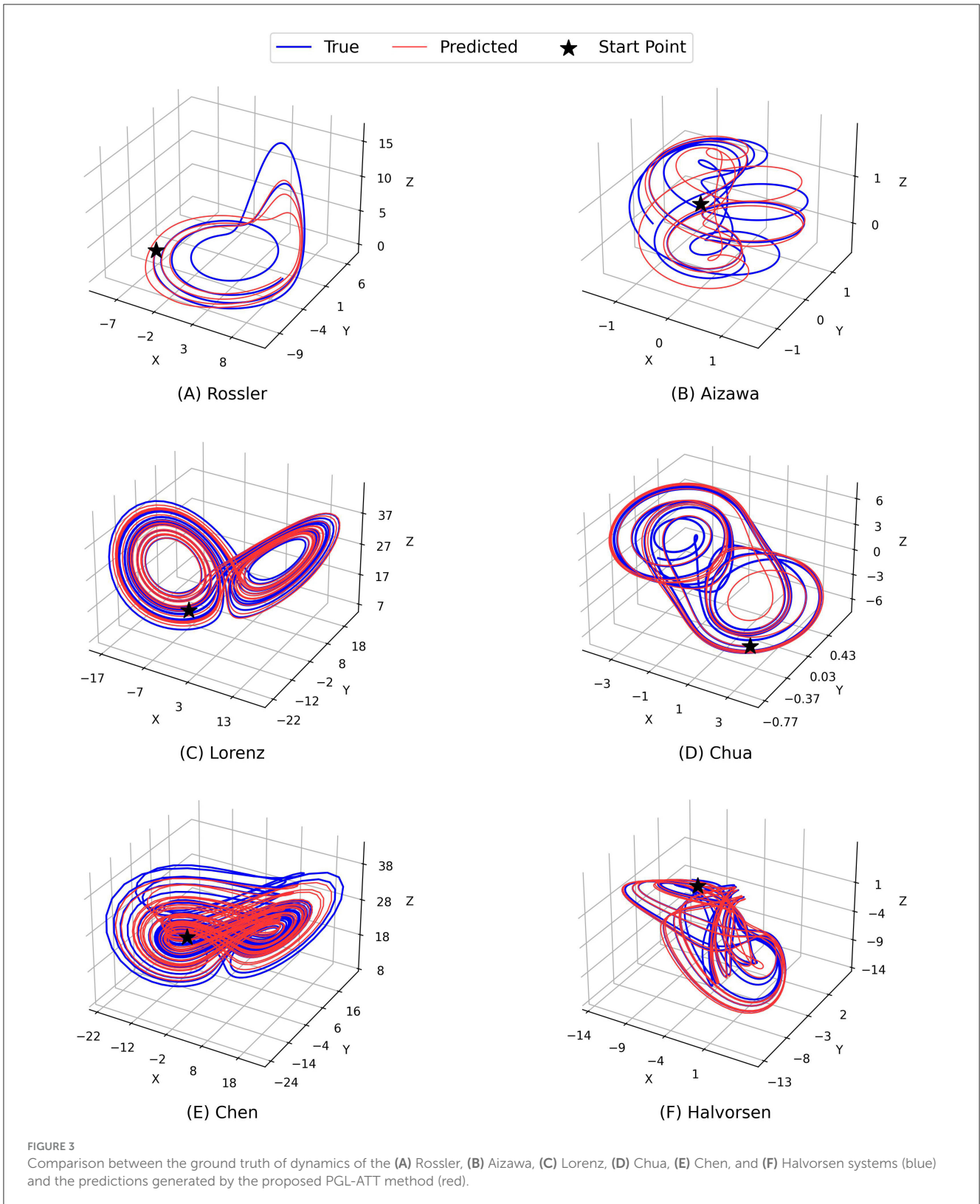
## 3.3 Analysis of results

[Figures 2](#) and [3](#) demonstrate the comparison between the ground truth of dynamics of the Rossler, Aizawa, Lorenz, Chua, Chen, and Halvorsen systems in 2,000 time steps, which is illustrated in blue in each sub-figure, and the predictions generated by the proposed PGL-MLP ([Figure 2](#)) and PGL-ATT



(Figure 3) methods, which are shown in red. From these two figures, we can observe that both PGL-MLP and PGL-ATT can capture the dynamical patterns of these six chaotic systems. Although employing an iterative prediction process in the

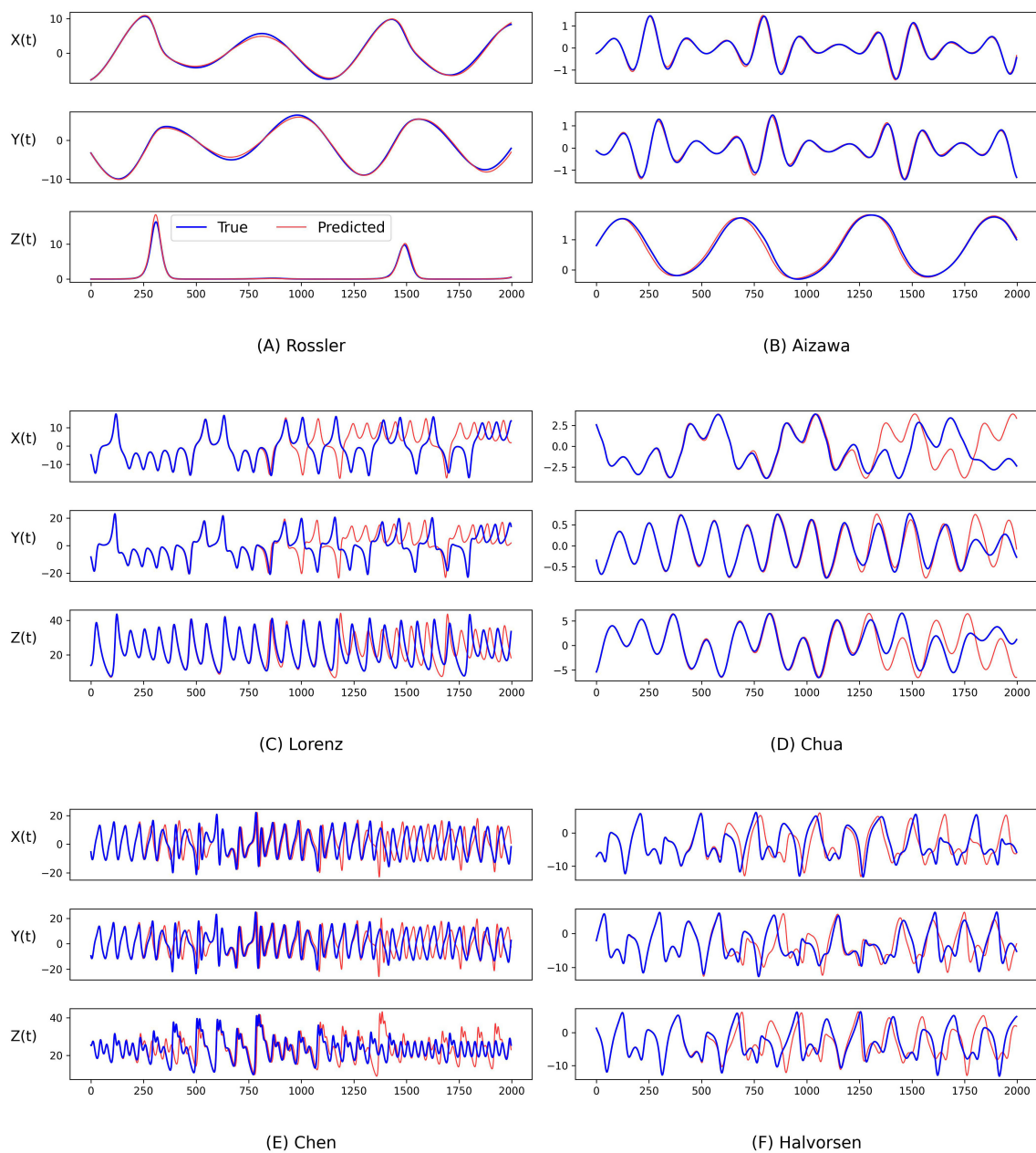
prediction phase brings great challenges to the task of long-term forecasting, the integration of data and physics enables our method to produce predictions that are consistent with actual dynamics.



To further evaluate the performance of our predictions, we also conduct an analysis by visualizing the temporal evolution of the ground truth and predictions of the state variables in these chaotic systems in Figures 4 and 5. Generally, both

PGL-MLP and PGL-ATT can make satisfactory predictions of the state variables  $X(t)$ ,  $Y(t)$ , and  $Z(t)$  for these chaotic systems. However, the performance of each method on different systems varies slightly. For the Rossler system, the predicted curves

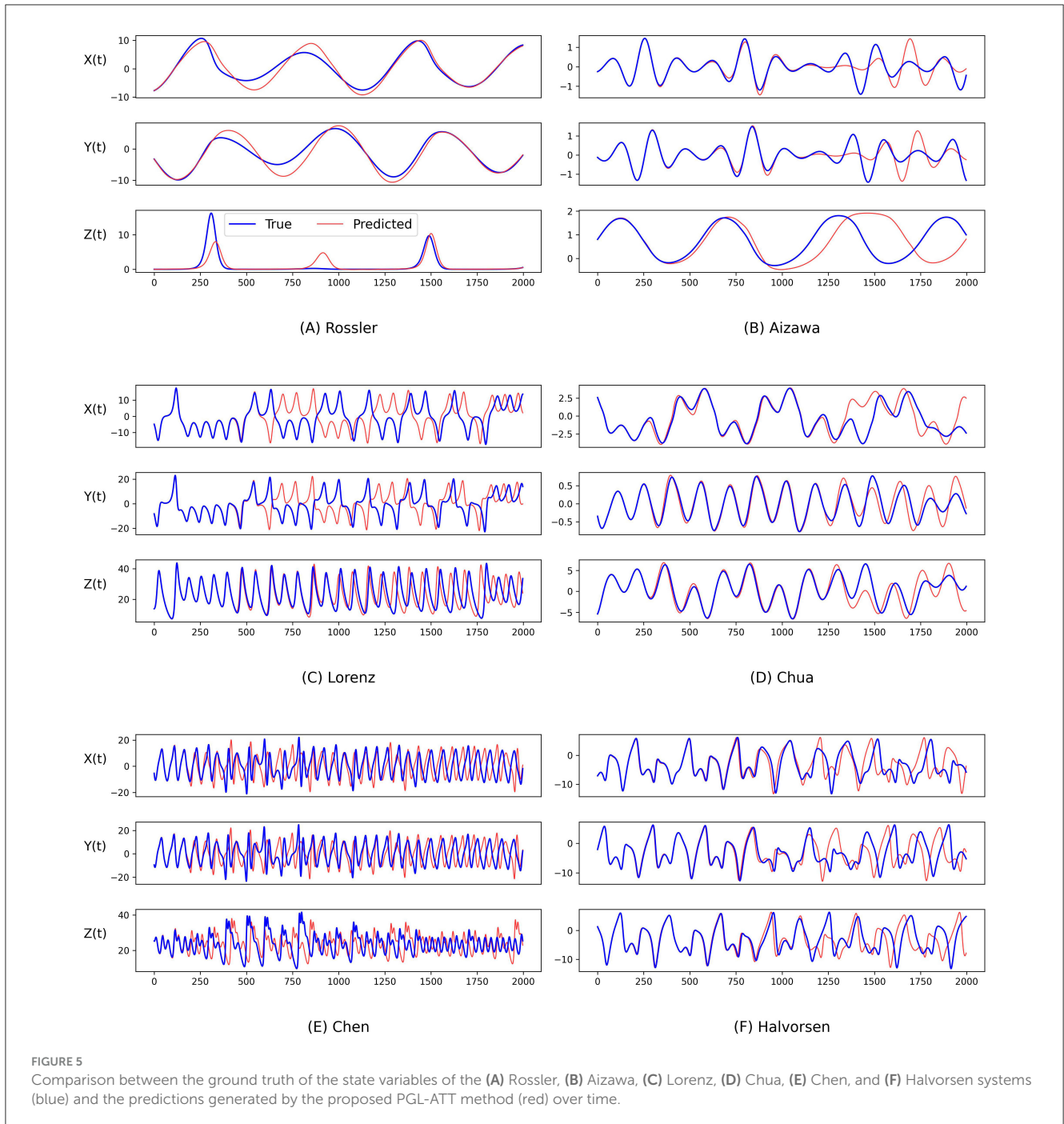




**FIGURE 4** Comparison between the ground truth of the state variables of the (A) Rossler, (B) Aizawa, (C) Lorenz, (D) Chua, (E) Chen, and (F) Halvorsen systems (blue) and the predictions generated by the proposed PGL-MLP method (red) over time.

of both PGL-MLP and PGL-ATT closely match the ground truth, accurately characterizing even the irregular patterns in  $Z(t)$  component; only one peak was missed by the PGL-ATT. This indicates that the proposed method successfully captures the dynamics of this chaotic system and thus is able to make accurate predictions in such a long-term period. For the Aizawa system, the PGL-MLP shows very good performance; its prediction is consistent with the ground truth in all 2,000 steps. The performance of the PGL-ATT is also acceptable; the predicted dynamics match well with the actual curve in the first 1,000 steps. For the Lorenz system, both PGL-MLP

and PGL-ATT achieve high accuracy up to around 1,100 time steps on the component  $Z(t)$ , and 600 time steps on the components  $X(t)$  and  $Y(t)$ , respectively. For the Chua system, PGL-MLP and PGL-ATT have similar performance, making accurate predictions up to about 1250 time steps, and then exhibit notable discrepancies in the components  $X(t)$ ,  $Y(t)$ , and  $Z(t)$ . Such discrepancies in Chen and Halvorsen systems appear earlier, compared with the Chua system. Interestingly, PGL-MLP's predictions for both the Chen and Halvorsen systems initially achieve high accuracy but subsequently exhibit noticeable disturbances. Fortunately, due to the model's ability to balance data



and physical knowledge, it regains accuracy in its predictions after these disturbances.

To quantitatively compare the performance of our methods (i.e., PGL-MLP and PGL-ATT) with that of existing methods, we report the MAE and RMSE of all methods for different prediction horizons in Tables 2, 3, respectively. The results demonstrate that the proposed methods achieve the lowest prediction errors in most of the settings, demonstrating the effectiveness of our methods in making long-term predictions of chaotic system dynamics. An interesting observation is that the performance of PGL-MLP is generally better than that of PGL-ATT, despite

the latter employing a more sophisticated attention mechanism. One potential explanation is that the complexity of the attention mechanism may lead to overfitting in the predictive model when compared to PGL-MLP. It is important to note that the task of predicting chaotic system dynamics differs from natural language processing, where attention mechanisms have demonstrated notable effectiveness. The former focuses on capturing the intrinsic, evolving patterns of dynamical systems, which may change over time, whereas the latter is primarily concerned with understanding consistent contextual relationships in input data. Consequently, a model that is overly complex or overfitted to historical data

TABLE 2 MAE of LSTM, ESN, NG-RC, DLinear, K-NODE, and the proposed PGL in different prediction horizons on six chaotic systems.

Systems	Horizon	LSTM	ESN	NG-RC	DLinear	K-NODE	PGL-ATT	PGL-MLP
Rossler	200-horizon	0.295	0.228	0.046	2.374	<b>0.039</b>	0.206	0.055
	600-horizon	0.934	1.411	0.865	4.318	0.322	1.264	<b>0.182</b>
	1,000-horizon	0.705	3.126	1.084	4.625	0.409	1.539	<b>0.236</b>
	1,400-horizon	0.583	3.919	1.199	5.111	0.481	1.341	<b>0.208</b>
	2,000-horizon	0.744	4.713	1.942	5.341	0.795	1.044	<b>0.225</b>
Aizawa	200-horizon	0.142	0.396	0.205	1.249	0.022	<b>0.011</b>	0.028
	600-horizon	0.198	0.649	0.419	1.104	0.034	<b>0.031</b>	0.049
	1,000-horizon	0.337	0.765	0.455	1.188	<b>0.058</b>	0.068	0.067
	1,400-horizon	0.444	0.807	0.497	1.292	0.103	0.137	<b>0.061</b>
	2,000-horizon	0.529	0.817	0.485	1.456	0.133	0.320	<b>0.060</b>
Lorenz	200-horizon	2.668	0.631	1.105	7.748	1.020	<b>0.245</b>	0.338
	600-horizon	4.828	5.462	5.088	6.602	3.618	<b>0.559</b>	0.616
	1,000-horizon	5.996	7.103	4.287	7.316	4.828	4.413	<b>1.189</b>
	1,400-horizon	6.579	8.110	5.285	7.541	5.964	5.915	<b>3.685</b>
	2,000-horizon	7.103	8.174	6.774	7.984	7.169	7.117	<b>5.536</b>
Chua	200-horizon	0.163	0.104	1.057	1.007	<b>0.008</b>	0.035	0.023
	600-horizon	1.372	0.999	1.815	1.321	0.095	0.253	<b>0.079</b>
	1,000-horizon	1.382	1.417	1.967	1.568	0.563	0.244	<b>0.117</b>
	1,400-horizon	1.242	1.855	2.047	1.622	1.042	0.321	<b>0.285</b>
	2,000-horizon	1.281	2.187	1.948	1.620	1.535	0.714	<b>0.912</b>
Chen	200-horizon	5.531	3.093	3.770	4.876	0.675	1.033	<b>0.300</b>
	600-horizon	8.085	6.480	7.906	6.182	6.017	5.936	<b>4.439</b>
	1,000-horizon	7.349	7.859	8.888	6.576	7.441	7.268	<b>4.526</b>
	1,400-horizon	8.026	8.323	8.731	6.604	7.682	7.152	<b>5.146</b>
	2,000-horizon	7.996	8.012	9.166	<b>6.484</b>	8.263	8.039	6.535
Halvorsen	200-horizon	0.456	1.168	0.434	2.167	0.063	<b>0.044</b>	0.107
	600-horizon	1.743	3.937	2.348	3.453	1.635	<b>0.217</b>	3.343
	1,000-horizon	1.515	5.155	2.872	3.575	1.241	1.991	<b>0.819</b>
	1,400-horizon	2.146	6.040	3.092	3.564	<b>1.297</b>	2.093	1.329
	2,000-horizon	3.207	6.642	3.655	3.610	<b>1.769</b>	2.486	2.550

The best performance in each setting is highlighted in bold.

may not yield the expected performance in predicting chaotic systems dynamics.

In addition to the MAE and RMSE, we further analyze the performance of the comparison baselines and our proposed methods using the  $R^2$  metric, which ranges from 0 to 1 to indicate performance quality. As illustrated in Figure 6, we plot the  $R^2$  score's trend with increasing predicted time steps and calculate the specific Lyapunov Time for different forecasting horizons. The Lyapunov Time is a critical indicator of a system's chaotic behavior, representing the duration over which two initially close trajectories will diverge significantly (Sangiorgio and Dercole, 2020; Sangiorgio et al., 2021, 2022; Pathak et al., 2018; Patel et al., 2021; Vlachas et al., 2020). Our results show that the proposed methods achieve improved performance across the six chaotic systems. However, the

performance of all methods varies across different chaotic systems. This variability is likely due to each system's unique Lyapunov Time, presenting different levels of prediction difficulty.

### 3.4 Ablation study

In this subsection, we conduct an ablation study to understand the individual contributions of the different components within our proposed method to predict chaotic dynamics. Specifically, we examine the performance of the Lorenz system dynamics prediction using four distinct configurations of our method: (1) employing only the DDC, which is an LSTM network;

TABLE 3 RMSE of LSTM, ESN, NG-RC, DLinear, K-NODE, and the proposed PGL-ATT and PGL-MLP in different prediction horizons on six chaotic systems.

Systems	Horizon	LSTM	ESN	NG-RC	DLinear	K-NODE	PGL-ATT	PGL-MLP
Rossler	200-horizon	0.354	0.277	0.053	3.103	<b>0.046</b>	0.279	0.064
	600-horizon	1.692	2.435	1.483	5.370	0.546	2.117	<b>0.315</b>
	1,000-horizon	1.371	5.050	1.637	5.562	0.610	2.295	<b>0.373</b>
	1,400-horizon	1.186	5.789	1.723	5.963	0.690	2.011	<b>0.331</b>
	2,000-horizon	1.352	6.670	3.209	6.133	1.318	1.175	<b>0.341</b>
Aizawa	200-horizon	0.187	0.508	0.242	1.597	0.029	<b>0.014</b>	0.043
	600-horizon	0.245	0.758	0.525	1.361	<b>0.044</b>	0.045	0.064
	1,000-horizon	0.451	0.907	0.563	1.416	<b>0.077</b>	0.099	0.088
	1,400-horizon	0.587	0.964	0.620	1.543	0.164	0.238	<b>0.081</b>
	2,000-horizon	0.673	0.975	0.600	1.741	0.202	0.513	<b>0.079</b>
Lorenz	200-horizon	4.511	0.922	1.624	9.884	1.733	<b>0.380</b>	0.561
	600-horizon	7.834	8.372	8.693	8.550	7.113	1.101	<b>0.891</b>
	1,000-horizon	8.900	9.667	7.773	9.410	8.131	8.266	<b>2.864</b>
	1,400-horizon	9.401	10.560	8.689	9.638	9.214	9.799	<b>7.405</b>
	2,000-horizon	9.967	10.595	10.067	10.050	10.15	10.632	<b>8.982</b>
Chua	200-horizon	0.209	0.123	1.398	1.274	<b>0.010</b>	0.040	0.027
	600-horizon	2.021	1.617	2.274	1.569	0.142	0.372	<b>0.111</b>
	1,000-horizon	1.992	2.038	2.402	1.854	1.162	0.339	<b>0.162</b>
	1,400-horizon	1.800	2.515	2.468	1.919	1.753	<b>0.503</b>	0.535
	2,000-horizon	1.795	2.855	2.350	1.918	2.264	<b>1.174</b>	1.495
Chen	200-horizon	8.174	5.487	5.591	6.229	1.071	1.536	<b>0.483</b>
	600-horizon	10.476	8.944	9.890	7.649	8.975	8.945	<b>7.298</b>
	1,000-horizon	9.738	10.251	10.824	8.165	10.134	9.915	<b>6.953</b>
	1,400-horizon	10.448	10.593	10.778	8.138	10.093	9.568	<b>7.680</b>
	2,000-horizon	10.151	10.322	11.147	<b>7.910</b>	10.641	10.335	9.124
Halvorsen	200-horizon	0.600	1.966	0.633	2.761	0.102	<b>0.064</b>	0.142
	600-horizon	2.686	5.912	3.323	4.342	2.822	<b>0.389</b>	0.529
	1,000-horizon	2.283	7.207	3.785	4.418	2.289	3.624	<b>1.427</b>
	1,400-horizon	3.142	7.999	3.953	4.414	2.234	3.449	<b>2.260</b>
	2,000-horizon	4.573	8.431	4.638	4.429	2.698	<b>3.666</b>	4.019

The best performance in each setting is highlighted in bold.

(2) integrating both DDC and PGC through a simple linear combination, referred to as PGL-Linear; (3) implementing the proposed method with attention-based NLC as described in this manuscript, referred to as PGL-ATT; and (4) implementing the proposed method with MLP-based NLC as described in this manuscript, referred to as PGL-MLP. In this ablation study, all experimental settings remain consistent with those used in previous experiments, including the initial conditions, the ratio of training and testing sets, and the prediction horizons.

Table 4 presents the results of the ablation study with respect to MAE and RMSE across various forecast horizons. The results obtained from the DDC alone exhibit relatively high MAE and RMSE across all prediction horizons. When integrating the DDC with the PGC using a simple linear

combination (denoted as PGL-Linear), there is an observable improvement in performance compared to the DDC results. However, the enhancement achieved by PGL-Linear falls short of our expectations. One potential reason for this is that the relationship between the observational data and the physical principles governing the system's dynamics is likely nonlinear. As a result, a straightforward linear combination may be insufficient to capture the complexity of these interactions. This highlights the necessity of the proposed nonlinear combination (NLC) design for effectively integrating the DDC and PGC to enhance prediction accuracy. This necessity is further supported by the results from PGL-ATT and PGL-MLP, which demonstrate improved performance in terms of MAE and RMSE across all prediction horizons.

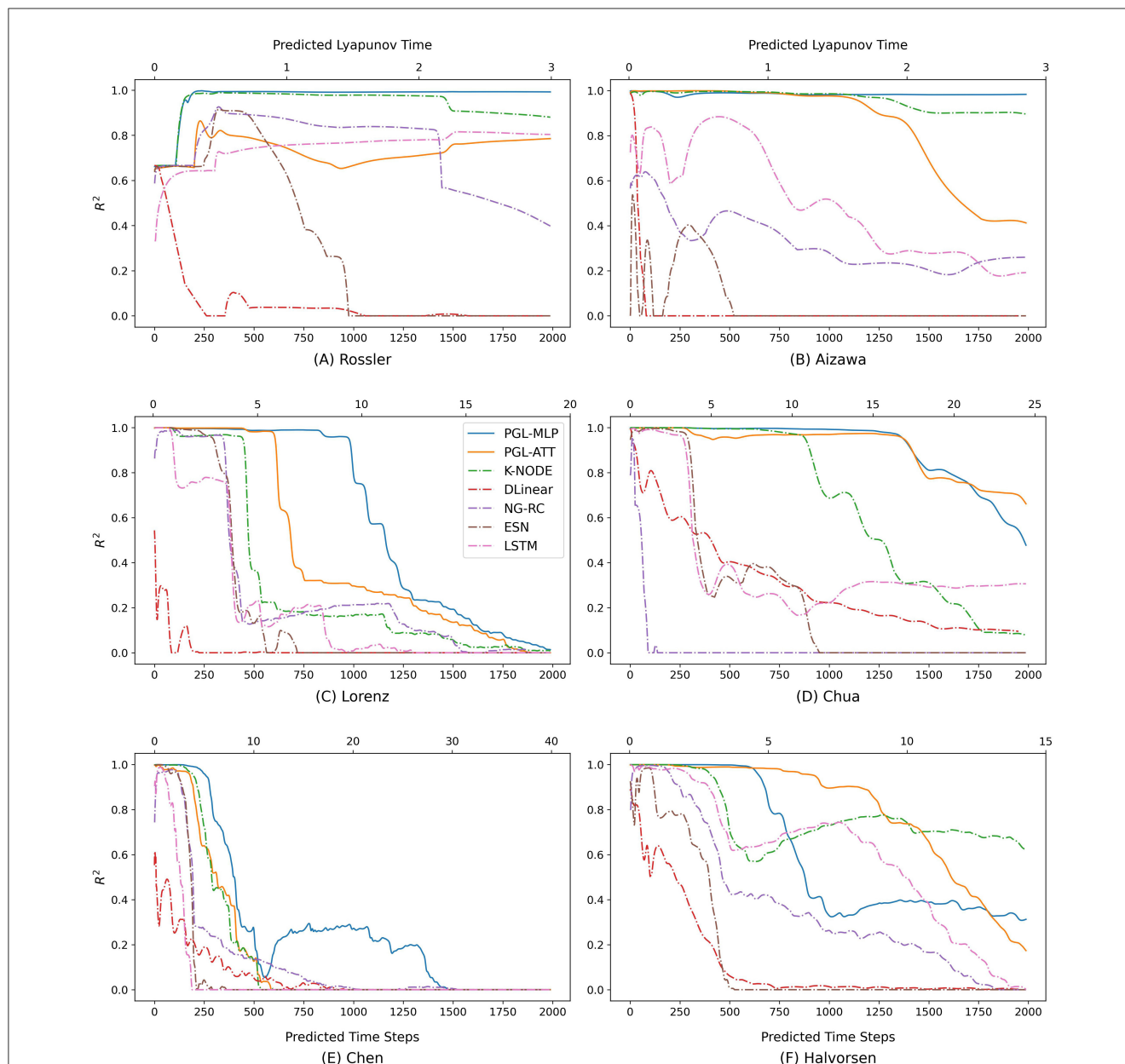


FIGURE 6

$R^2$  of LSTM, ESN, NG-RC, DLinear, K-NODE, and the proposed PGL-ATT and PGL-MLP in different prediction horizons on six chaotic systems. (A) Rossler, (B) Aizawa, (C) Lorenz, (D) Chua, (E) Chen, and (F) Halvorsen.

## 4 Conclusion and discussion

In this paper, we proposed a physics-guided learning approach to predict the dynamics of chaotic systems. We experimentally evaluated the performance of our method on the Rossler, Aizawa, Lorenz, Chua, Chen, and Halvorsen dynamical systems. The experimental results demonstrated that our method outperforms other baselines in terms of prediction accuracy.

To our knowledge, PINN is among several representative techniques that employ neural networks to solve ordinary and partial differential equations. Other noteworthy methods include those based on the Deep Galerkin Method (DGM) (Sirignano and Spiliopoulos, 2018; Aristotelous et al., 2023) and Neurodifferential

approaches (Lagaris et al., 1998; Ramuhalli et al., 2005), each offering unique contributions to the field. In our work, we utilize PINN as a typical example to demonstrate the efficacy of integrating data-driven structures with physical knowledge to accurately predict the dynamics of chaotic systems. This exemplification paves the way for further exploration into the integration of other physics-guided modules with data-driven components, potentially leading to enhanced predictive capabilities.

In our future work, we aim to extend our framework to scenarios where observations are noisy and the underlying governing differential equations are not fully known in advance. Moreover, in our current study, we used only six representative chaotic systems that exhibit distinct dynamical patterns such

TABLE 4 MAE and RMSE of DDC, PGL-Linear, PGL-ATT, and PGL-MLP in different prediction horizons on the Lorenz system.

Metrics	Horizon	DDC	PGL-Linear	PGL-ATT	PGL-MLP
MAE	200-horizon	2.668	1.691	<b>0.245</b>	0.338
	600-horizon	4.828	3.389	<b>0.559</b>	0.616
	1,000-horizon	5.996	4.619	4.413	<b>1.189</b>
	1,400-horizon	6.579	5.506	5.915	<b>3.685</b>
	2,000-horizon	7.103	6.839	7.117	<b>5.536</b>
RMSE	200-horizon	4.511	2.823	<b>0.380</b>	0.561
	600-horizon	7.834	5.829	1.101	<b>0.891</b>
	1,000-horizon	8.900	7.540	8.266	<b>2.864</b>
	1,400-horizon	9.401	8.395	9.799	<b>7.405</b>
	2,000-horizon	9.967	9.493	10.632	<b>8.892</b>

The best performance in each setting is highlighted in bold.

as the spiral-type chaos in the Rossler system (Rössler, 1977), the butterfly-shaped pattern in the Lorenz system (Li and Yin, 2009), and the double-scroll attractor in the Chua system (Chua, 2007) to demonstrate the feasibility of the proposed idea. Moving forward, we plan to conduct more comprehensive tests on 131 diverse chaotic systems across various domains (Gilpin, 2021) to further validate the robustness of our learning framework. Further, we intend to apply the proposed method to various real-world applications, such as infectious disease risk prediction, climate forecast, and traffic flow prediction. Additionally, we plan to conduct a comprehensive theoretical analysis of the proposed learning framework, attempting to quantitatively characterize its learning capacity and prediction error bounds using a series of key properties of chaotic systems, such as the Lyapunov Exponent and the Hurst Exponent.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

LF: Conceptualization, Data curation, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. YL: Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing. BS: Funding acquisition, Investigation, Methodology, Writing – original draft, Writing – review & editing. JL: Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work

was supported in part by the Ministry of Science and Technology of China (2021ZD0112500), in part by the National Natural Science Foundation of China and the Research Grants Council (RGC) of Hong Kong Joint Research Scheme (No. 62261160387, N\_HKBU222/22), in part by the Hong Kong Research Grants Council General Research Fund (RGC/HKBU12202220, RGC/HKBU12203122, and RGC/HKBU12200124), and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant no. SJCX23\_0435).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Generative AI statement

The author(s) declare that Gen AI was used in the creation of this manuscript. We acknowledge the use of ChatGPT (GPT-4, OpenAI's language model: <http://openai.com>) in polishing some of the wordings in the manuscript.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fdata.2024.1506443/full#supplementary-material>

## References

- Aguiar, M., Kooi, B., and Stollenwerk, N. (2008). Epidemiology of dengue fever: a model with temporary cross-immunity and possible secondary infection shows bifurcations and chaotic behaviour in wide parameter regions. *Math. Model. Nat. Phenom.* 3, 48–70. doi: 10.1051/mmnp:2008070
- Aizawa, Y., and Uezu, T. (1982). Topological aspects in chaos and in 2-k period doubling cascade. *Prog. Theoret. Phys.* 67, 982–985. doi: 10.1143/PTP.67.982
- Amaranto, A., and Mazzoleni, M. (2023). B-ama: A python-coded protocol to enhance the application of data-driven models in hydrology. *Environm. Model. Softw.* 160:105609. doi: 10.1016/j.envsoft.2022.105609
- Aristotelous, A. C., Mitchell, E. C., and Maroulas, V. (2023). Adlgm: an efficient adaptive sampling deep learning galerkin method. *J. Comput. Phys.* 477:111944. doi: 10.1016/j.jcp.2023.111944
- Chantry, M., Christensen, H., Dueben, P., and Palmer, T. (2021). Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft ai. *Philosoph. Trans. Royal Soc. A* 379:20200083. doi: 10.1098/rsta.2020.0083
- Chattopadhyay, A., Hassanzadeh, P., and Subramanian, D. (2020). Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process. Geophys.* 27, 373–389. doi: 10.5194/npg-27-373-2020
- Chen, G., and Ueta, T. (1999). Yet another chaotic attractor. *Int. J. Bifurcat. Chaos* 9, 1465–1466. doi: 10.1142/S0218127499001024
- Cheng, W., Wang, Y., Peng, Z., Ren, X., Shuai, Y., Zang, S., et al. (2021). High-efficiency chaotic time series prediction based on time convolution neural network. *Chaos, Solit. Fractals* 152:111304. doi: 10.1016/j.chaos.2021.111304
- Chua, L. (1971). Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* 18, 507–519. doi: 10.1109/TCT.1971.1083337
- Chua, L., Komuro, M., and Matsumoto, T. (1986). The double scroll family. *IEEE Trans. Circuits Syst.* 33, 1072–1118. doi: 10.1109/TCS.1986.1085869
- Chua, L. O. (2007). Chua circuit. *Scholarpedia* 2:1488. doi: 10.4249/scholarpedia.1488
- Deng, Q., Wang, C., and Lin, H. (2024). Chaotic dynamical system of hopfield neural network influenced by neuron activation threshold and its image encryption. *Nonlinear Dyn.* 112, 6629–6646. doi: 10.1007/s11071-024-09384-3
- Doan, N. A. K., Polifke, W., and Magri, L. (2020). Physics-informed echo state networks. *J. Comput. Sci.* 47:101237. doi: 10.1016/j.jocs.2020.101237
- Feng, L., Liu, Y., Shi, B., and Liu, J. (2024). “Learning dynamical systems from data: an introduction to physics-guided deep learning,” in *Proceedings of the 4th French Regional Conference on Complex Systems* (Springer Nature and Frontiers), 57–71.
- Field, R. J., and Gyorgyi, L. (1993). *Chaos in Chemistry and Biochemistry*. Singapore: World Scientific.
- Gao, H., Huang, B., Chen, G., Xia, L., and Radenkovic, M. (2024). Deep learning solver unites sdgsat-1 observations and navier-stokes theory for oceanic vortex streets. *Remote Sens. Environ.* 315:114425. doi: 10.1016/j.rse.2024.114425
- Gauthier, D. J., Bollt, E., Griffith, A., and Barbosa, W. A. (2021). Next generation reservoir computing. *Nat. Commun.* 12, 1–8. doi: 10.1038/s41467-021-25801-2
- Gilpin, W. (2021). “Chaos as an interpretable benchmark for forecasting and data-driven modelling,” in *NeurIPS 2021 Datasets and Benchmarks Track (Round 2)* (Curran Associates).
- Grassberger, P., and Procaccia, I. (1983). Characterization of strange attractors. *Phys. Rev. Lett.* 50:346. doi: 10.1103/PhysRevLett.50.346
- Hess, B. (1990). Order and chaos in chemistry and biology. *Fresenius' J. Analyt. Chem.* 337, 459–468. doi: 10.1007/BF00322848
- Hochreiter, S. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735
- Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci.* 81, 3088–3092. doi: 10.1073/pnas.81.10.3088
- Hu, M., Huang, X., Shi, Q., Yuan, F., and Wang, Z. (2024). Design and analysis of a memristive hopfield switching neural network and application to privacy protection. *Nonlinear Dynam.* 112, 12485–12505. doi: 10.1007/s11071-024-09696-4
- Jaeger, H. (2001). *The “Echo State” Approach to Analysing and Training Recurrent Neural Networks-With an Erratum Note*. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, 13.
- Jiahao, T. Z., Hsieh, M. A., and Forgoston, E. (2021). Knowledge-based learning of nonlinear dynamics and chaos. *Chaos* 31:11. doi: 10.1063/5.0065617
- Kennedy, C., Crowdis, T., Hu, H., Vaidyanathan, S., and Zhang, H.-K. (2024). Data-driven learning of chaotic dynamical systems using discrete-temporal sobolev networks. *Neural Netw* 173:106152. doi: 10.1016/j.neunet.2024.106152
- Kong, X., Yu, F., Yao, W., Cai, S., Zhang, J., and Lin, H. (2024). Memristor-induced hyperchaos, multiscroll and extreme multistability in fractional-order hnn: Image encryption and fpga implementation. *Neural Netw* 171, 85–103. doi: 10.1016/j.neunet.2023.12.008
- Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* 9, 987–1000. doi: 10.1109/72.712178
- Li, D., and Yin, Z. (2009). Connecting the lorenz and chen systems via nonlinear control. *Commun. Nonlinear Sci. Numer. Simulat.* 14, 655–667. doi: 10.1016/j.cnsns.2007.11.012
- Li, R., Dong, E., Tong, J., and Wang, Z. (2022). A novel multiscroll memristive hopfield neural network. *Int. J. Bifurcat. Chaos* 32:2250130. doi: 10.1142/S0218127422501309
- Liu, L., Zhang, L., Jiang, D., Guan, Y., and Zhang, Z. (2019). A simultaneous scrambling and diffusion color image encryption algorithm based on hopfield chaotic neural network. *IEEE Access* 7, 185796–185810. doi: 10.1109/ACCESS.2019.2961164
- Lorenz, E. N. (1963). Deterministic nonperiodic flow. *J. Atmosph. Sci.* 20, 130–141.
- Mangiarotti, S., Peyre, M., and Huc, M. (2016). A chaotic model for the epidemic of ebola virus disease in west africa (2013–2016). *Chaos* 26:113112. doi: 10.1063/1.4967730
- Mishra, A., Purohit, S., Owolabi, K., and Sharma, Y. (2020). A nonlinear epidemiological model considering asymptotic and quarantine classes for sars cov-2 virus. *Chaos, Solitons Fractals* 138:109953. doi: 10.1016/j.chaos.2020.109953
- Misyris, G. S., Venzke, A., and Chatzivasileiadis, S. (2020). “Physics-informed neural networks for power systems,” in *2020 IEEE Power & Energy Society General Meeting (PESGM)* (Montreal, QC: IEEE), 1–5.
- Na, X., Li, Y., Ren, W., and Han, M. (2023). Physics-informed hierarchical echo state network for predicting the dynamics of chaotic systems. *Expert Syst. Appl.* 228:120155. doi: 10.1016/j.eswa.2023.120155
- Na, X., Ren, W., and Xu, X. (2021). Hierarchical delay-memory echo state network: a model designed for multi-step chaotic time series prediction. *Eng. Appl. Artif. Intell.* 102:104229. doi: 10.1016/j.engappai.2021.104229
- Nasiri, H., and Ebadzadeh, M. M. (2022). MFRFNN: multi-functional recurrent fuzzy neural network for chaotic time series prediction. *Neurocomputing* 507, 292–310. doi: 10.1016/j.neucom.2022.08.032
- Nath, K., Meng, X., Smith, D. J., and Karniadakis, G. E. (2023). Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines. *Sci. Rep.* 13:13683. doi: 10.1038/s41598-023-39989-4
- Ning, X., Guan, J., Li, X.-A., Wei, Y., and Chen, F. (2023). Physics-informed neural networks integrating compartmental model for analyzing COVID-19 transmission dynamics. *Viruses* 15:1749. doi: 10.3390/v15081749
- Olsen, P. E., Laskar, J., Kent, D. V., Kinney, S. T., Reynolds, D. J., Sha, J., et al. (2019). Mapping solar system chaos with the geological orrery. *Proc. Nat. Acad. Sci.* 116, 10664–10673. doi: 10.1073/pnas.1813901116
- Özalp, E., Margazoglou, G., and Magri, L. (2023). “Physics-informed long short-term memory for forecasting and reconstruction of chaos,” in *International Conference on Computational Science* (Springer), 382–389.
- Palmer, T. N. (1993). Extended-range atmospheric prediction and the lorenz model. *Bull. Am. Meteorol. Soc.* 74, 49–66.2.0.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). “Automatic differentiation in pytorch,” in *NeurIPS Autodiff Workshop* (Curran Associates).
- Patel, D., Canaday, D., Girvan, M., Pomerance, A., and Ott, E. (2021). Using machine learning to predict statistical properties of non-stationary dynamical processes: system climate, regime transitions, and the effect of stochasticity. *Chaos* 31:42598. doi: 10.1063/5.0042598
- Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. (2018). Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* 120:024102. doi: 10.1103/PhysRevLett.120.024102
- Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E. (2017). Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos* 27:12. doi: 10.1063/1.5010300
- Pecora, L. M., and Carroll, T. L. (1990). Synchronization in chaotic systems. *Phys. Rev. Lett.* 64:821. doi: 10.1103/PhysRevLett.64.821
- Racca, A., and Magri, L. (2021). “Automatic-differentiated physics-informed echo state network (API-ESN),” in *International Conference on Computational Science* (Cham: Springer), 323–329.
- Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378:686–707. doi: 10.1016/j.jcp.2018.10.045

- Ramuhalli, P., Udpa, L., and Udpa, S. S. (2005). Finite-element neural networks for solving differential equations. *IEEE Trans. Neural Netw.* 16, 1381–1392. doi: 10.1109/TNN.2005.857945
- Rössler, O. E. (1976). An equation for continuous chaos. *Physics Letters A* 57, 397–398. doi: 10.1016/0375-9601(76)90101-8
- Rössler, O. E. (1977). Chaos in abstract kinetics: two prototypes. *Bull. Math. Biol.* 39, 275–289. doi: 10.1016/S0092-8240(77)80015-3
- Sangiorgio, M., and Dercole, F. (2020). Robustness of lstm neural networks for multi-step forecasting of chaotic time series. *Chaos, Solitons Fract.* 139:110045. doi: 10.1016/j.chaos.2020.110045
- Sangiorgio, M., Dercole, F., and Guariso, G. (2021). Forecasting of noisy chaotic systems with deep neural networks. *Chaos, Solitons Fract.* 153:111570. doi: 10.1016/j.chaos.2021.111570
- Sangiorgio, M., Dercole, F., and Guariso, G. (2022). *Deep Learning in Multi-Step Prediction of Chaotic Dynamics: From Deterministic Models to Real-World Systems* (Cham: Springer).
- Shi, B., Feng, L., He, H., Hao, Y., Peng, Y., Liu, M., et al. (2024). A physics-guided attention-based neural network for sea surface temperature prediction. *IEEE Trans. Geosci. Remote Sensing* 62, 1–13. doi: 10.1109/TGRS.2024.3457039
- Sirignano, J., and Spiliopoulos, K. (2018). Dgm: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys.* 375, 1339–1364. doi: 10.1016/j.jcp.2018.08.029
- Sprott, J. C. (2010). *Elegant Chaos: Algebraically Simple Chaotic Flows*. Singapore: World Scientific.
- Toreti, A., Naveau, P., Zampieri, M., Schindler, A., Scoccimarro, E., Xoplaki, E., et al. (2013). Projections of global changes in precipitation extremes from coupled model intercomparison project phase 5 models. *Geophys. Res. Lett.* 40, 4887–4892. doi: 10.1002/grl.50940
- Vaswani, A. (2017). “Attention is all you need,” in *Proceedings of Advances in Neural Information Processing Systems* (Curran Associates).
- Vlachas, P.-R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., et al. (2020). Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Netw.* 126, 191–217. doi: 10.1016/j.neunet.2020.02.016
- Wu, G., Tang, L., and Liang, J. (2024). Synchronization of non-smooth chaotic systems via an improved reservoir computing. *Sci. Rep.* 14:229. doi: 10.1038/s41598-023-50690-4
- Yang, Q., Hernandez-Garcia, A., Harder, P., Ramesh, V., Sattgeger, P., Szwarcman, D., et al. (2023). Fourier neural operators for arbitrary resolution climate data downscaling. *arXiv [preprint]* arXiv:2305.14452. doi: 10.48550/arXiv.2305.14452
- Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2023). Are transformers effective for time series forecasting? *Proc. AAAI Conf. Artif. Intellig.* 37, 11121–11128. doi: 10.1609/aaai.v37i9.26317