mathematics

Article

Deep Embedding Koopman Neural Operator-Based Nonlinear Flight Training Trajectory Prediction Approach

Jing Lu *, Jingjun Jiang and Yidan Bai

College of Computer Science, Civil Aviation Flight University of China, Nanchang Road,
Guanghan 618307, China; jiangjingjun@cafuc.edu.cn (J.J.); baiyidan@cafuc.edu.cn (Y.B.)
* Correspondence: lujing_cafuc@nuaa.edu.cn

Abstract: Accurate flight training trajectory prediction is a key task in automatic flight maneuver evaluation and flight operations quality assurance (FOQA), which is crucial for pilot training and aviation safety management. The task is extremely challenging due to the nonlinear chaos of trajectories, the unconstrained airspace maps, and the randomization of driving patterns. In this work, a deep learning model based on data-driven modern koopman operator theory and dynamical system identification is proposed. The model does not require the manual selection of dictionaries and can automatically generate augmentation functions to achieve nonlinear trajectory space mapping. The model combines stacked neural networks to create a scalable depth approximator for approximating the finite-dimensional Koopman operator. In addition, the model uses finite-dimensional operator evolution to achieve end-to-end adaptive prediction. In particular, the model can gain some physical interpretability through operator visualization and generative dictionary functions, which can be used for downstream pattern recognition and anomaly detection tasks. Experiments show that the model performs well, particularly on flight training trajectory datasets.

Keywords: Koopman neural operator; nonlinear trajectories; flight trajectory prediction

MSC: 37M10

check for updates

Citation: Lu, J.; Jiang, J.; Bai, Y. Deep Embedding Koopman Neural Operator-Based Nonlinear Flight Training Trajectory Prediction Approach. *Mathematics* **2024**, *12*, 2162. https://doi.org/10.3390/math12142162

Academic Editors: Wei Fang and Victor S. Sheng

Received: 18 June 2024
Revised: 3 July 2024
Accepted: 6 July 2024
Published: 10 July 2024

1. Introduction

Trajectory data refer to spatial-temporal data generated by a moving object in geographic space [1,2]. They are typically represented by a series of spatial points with temporal order and their associated attributes. In the real world, trajectory data originate mainly from human activities, wild animals' migrations, transportation vehicle movements, and natural phenomena. Flight trajectories are categorized into flight transportation trajectories and flight training trajectories. Flight training trajectories are generated from the training of civil aviation pilots using fixed-wing aircraft, without considering the multi-target interaction problem. Currently, the assessment of a pilot's operational proficiency is typically carried out by instructors who observe the pilot's performance and assign scores based on flight regulations. This manual method of evaluation is relatively subjective and inefficient. Automated flight assessment technology can assist instructors in evaluating the core competencies of pilot training more efficiently, systematically, and comprehensively. The predictive analysis of flight trajectories is an integral part of this technology. It enables the automatic identification and assessment of the flight maneuvers that pilots are trained on, allowing for a more objective evaluation of the quality of a pilot's training exercises. Moreover, flight trajectory prediction plays a significant role in various aspects of flight safety, such as preventing mid-air collisions, responding to adverse weather conditions, planning for emergency responses, and optimizing flight paths. Consequently, the accurate prediction of flight training trajectories is a critical task for both automated flight maneuver assessment and Flight Operation Quality Assurance (FOQA), which are essential for pilot training and aviation safety management. The scientific content of this task involves flight trajectory data modeling [3] and anomaly identification [4]. However, due to the difficulty of the tasks and the high level of data confidentiality, there is little research disclosing related work. In contrast, there are many studies on vehicle transportation trajectories [5–7] and flight transportation trajectories. Research on flight transportation trajectories has focused on 4D trajectory prediction in the field of air traffic management (ATM) [8–13].

As shown in Figure 1, the flight training trajectory demonstrates the complexity of aircraft motion. The aircraft, with its six degrees of freedom and multiple coordinate systems, including the ground axis system, the airfares axis system, the trajectory axis system, and the airflow axis system [14], is not confined by the topology of the map. The pilot's decision-making process does not follow a fixed pattern and must consider safety conditions, making it more challenging than studying the trajectory of ground vehicles. Furthermore, flight trajectories are nonlinear due to the aerodynamic characteristics of the vehicle and engine thrust. Experiments [15] have confirmed the chaotic nature of these trajectories. Nonlinear trajectories are prevalent in nature [16], and their predictive modeling holds significant academic and practical value. This issue has been a focal point of research in the fields of human motion [17], trajectory tracking, and optimal control [18–21].

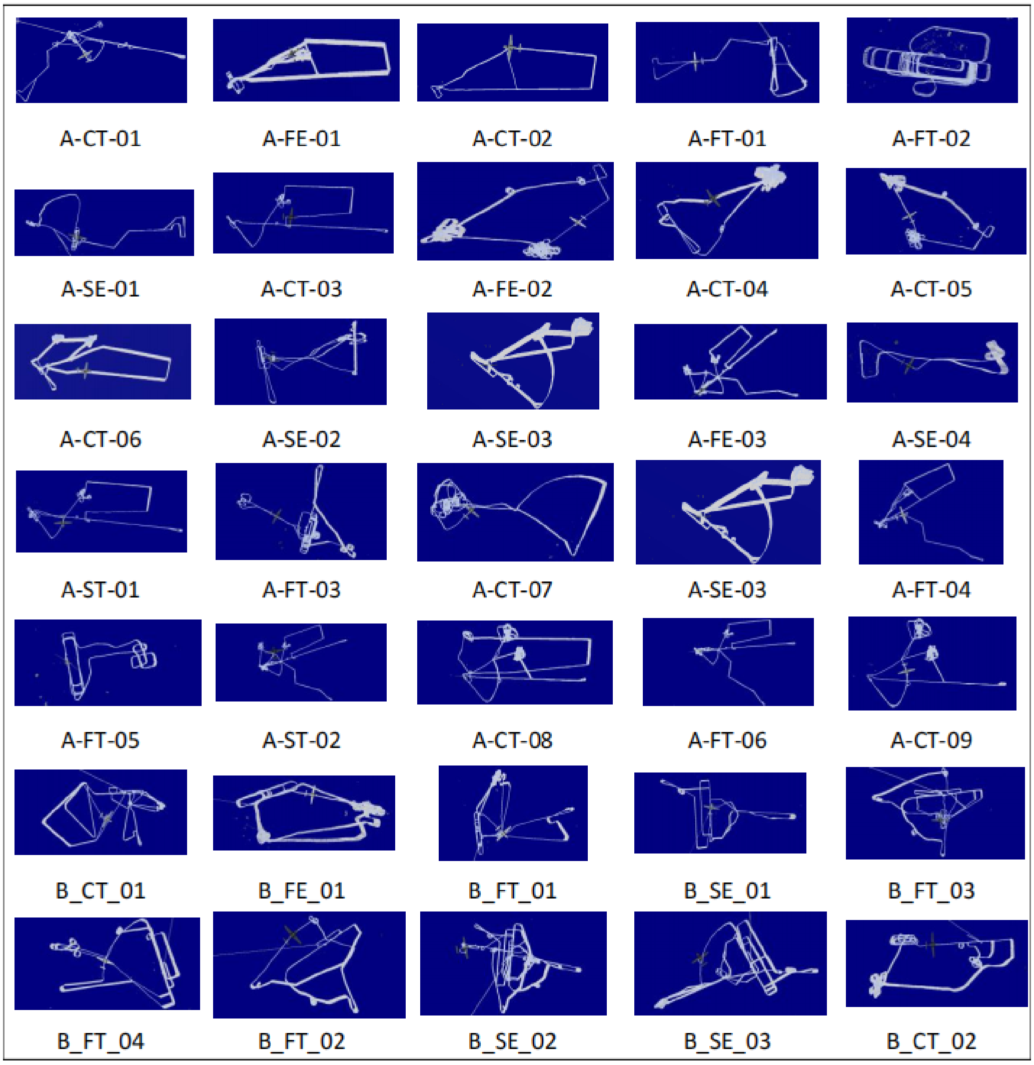


Figure 1. Flight training trajectories from the CAFUC dataset.

In 1931, Koopman proposed the theory of Koopman operators to address the issue of observables for Hamiltonian systems [22]. This theory was later expanded by Koopman and von Neumann in 1932 to encompass systems with continuous eigenvalue spectra [23].

This 1931 paper is the most famous central proof of the ergodic theorem formulated by von Neumann. It establishes a connection between Koopman's eigenvalue spectrum and conserved quantities, integrability, and ergodicity. The ergodic theorem is a branch of mathematics that examines the statistical properties of deterministic dynamical systems. It formally investigates the states of dynamical systems with invariant measures, such as the movements of the planets in the solar system. In this context, the planets move, but the rules governing their motions remain constant. In two papers published in 1932 [24,25], von Neumann made fundamental contributions to the theory of such systems, including von Neumann's mean ergodic theorem, which is considered to provide the first rigorous mathematical foundation for the statistical mechanics of liquids and gases.

Flight trajectory prediction has always been a key issue in air traffic management. In a paper by Zneg et al. [8], flight trajectory prediction methods were roughly divided into state estimation methods, dynamic methods, and machine learning methods. State estimation models are relatively simple. It only establishes equations based on the position, velocity, and acceleration of the aircraft, which will cause large errors and can only work for a short time. Dynamic methods analyze from the perspective of the aircraft, but most of them are implemented under the force state. Due to the large model parameters and the fact that the data are often limited, there will be large uncertainties and errors. Machine learning models do not need to display the modeling of the aircraft but only need to learn rules from massive amounts of data. These methods make it possible to mine complex trajectory patterns and extract important features because they can obtain a large amount of flight trajectory data. Different from the above methods, we propose a deep learning model based on data-driven modern Koopman operator theory and dynamic system identification.

Today, the Koopman operator has become an effective method for studying dynamical systems. Even strongly nonlinear systems, such as chaos systems, can be analyzed using the infinite-dimensional nonlinear Koopman operator, which is a linear observable function. Its spectral decomposition fully characterizes the behavior of nonlinear systems. Unlike the previous method of nonlinearization through local linearization, the Koopman theory demonstrates that the spectral characteristics of the linear operator can fully represent the characteristics of the nonlinear dynamical system. This allows for the creation of a global linearization model and the derivation of insightful analysis results using various established technical frameworks in linear analysis. In fact, in 2020, Lange et al. developed a nonlinear, data-driven model of a complex dynamical system using the Koopman theory. A nonlinear, data-driven spectral decomposition algorithm was validated for its effectiveness through predictive experiments on real-world power systems [26].

In addition, Fatemeh Daneshfar et al. [27] proposed a deep autoencoder with adaptive elastic loss (EDA-TEC) for text embedding clustering. Elahe Nasiri et al. [28] introduced an attribute embedding method to predict protein–protein interactions (PPI) in grids, improving prediction accuracy.

To address the nonlinear flight trajectory prediction problem, we developed a unified deep learning framework: Deep Embedding Koopman Neural Operator (DE-KNO) based on data-driven Koopman theory. This framework enables us to (1) construct and analyze models using only raw data obtained from observations, eliminating the need for manual feature function selection; and (2) accurately predict nonlinear flight trajectories and approximate neural operators for downstream tasks such as pattern recognition and anomaly detection in flight subjects. Our contributions are the following: (1) Examined the spatial transformation mapping of nonlinear trajectories and their properties in various spaces. (2) Implemented automatic trajectory modeling using a purely data-driven approach to avoid the manual selection of feature functions. (3) Attained some physical interpretability through generative physical models and visualization of neural operators. (4) Developed a unified deep learning framework for Koopman neural operators with generalization capabilities. (5) The proposed model performs outstandingly on the flight trajectory dataset compared to a state-of-the-art model.

2. Preliminaries

2.1. Koopman Operators

Koopman operator theory provides the dynamical equations for general discrete nonlinear systems:

$$x_{k+1} = f(x_k), x_k \in \mathbb{R}^n, f \in \mathcal{F} \tag{1}$$

where $k \in \mathbb{N}$ is the time step, f is the state-to-state mapping function in state space $\mathbb{R}^n$, $\mathcal{F}$ is the mapping function space, and x_k represents the state of the system.

Define Koopman operator $\mathcal{K}$ as a a linear operator acting on the observation function Θ of infinite dimensions and satisfying the following conditions:

$$\mathcal{K}\Theta(x_k) = \Theta(f(x_k)), \forall\Theta : \mathcal{H} \longrightarrow \mathbb{R} \tag{2}$$

That is, the Koopman operator acting on this set of observation functions is

$$\mathcal{K}\Theta = \Theta \circ \mathcal{F} \Rightarrow \mathcal{K}\Theta(x_k) = \Theta(x_{k+1}) \tag{3}$$

The above definition indicates that the Koopman operator is linear and infinite-dimensional. It only requires the use of the Koopman operator for direct matrix multiplication under the observation space to the state variable x_k for time recursion. In obtaining the observation function and the Koopman operator, a linear prediction of the state of the dynamical system can be achieved. This is more intuitively expressed in Figure 2 below.

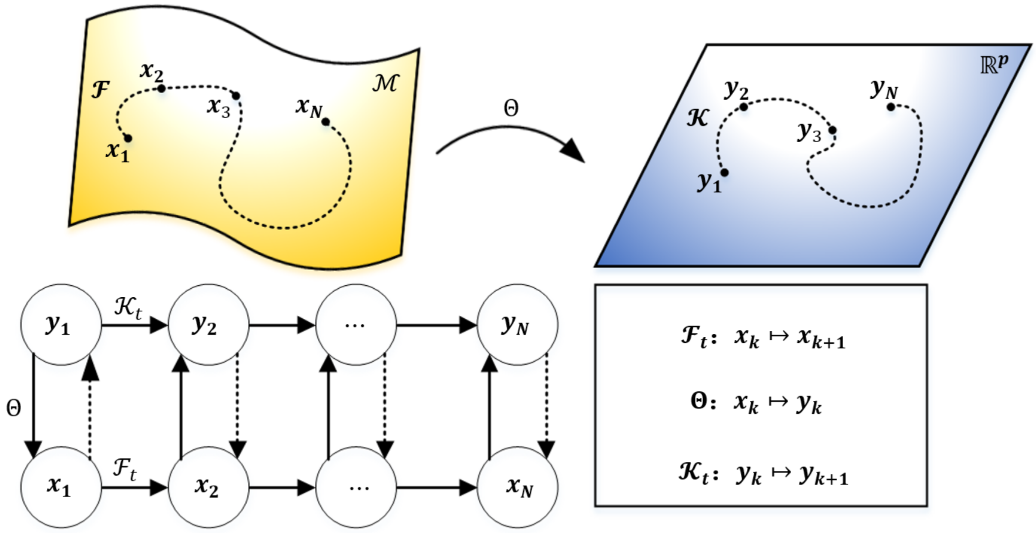


Figure 2. Koopman measure-invariant subspace.

This approach offers clear physical interpretability and relies solely on data-driven characteristics, compensating for the opaque nature of machine learning and deep learning. It also avoids the necessity for complex mathematical models. The essence of this approach is to transform the nonlinear system into an infinite-dimensional linear space, utilizing infinite dimensions to achieve linear properties. However, the practical implementation of infinite-dimensional computation poses challenges. Therefore, the primary focus of this method lies in the finite-dimensional approximation of the infinite-dimensional Koopman operator. The most commonly used methods for this problem are dynamic mode decomposition (DMD) [29] and extended dynamic mode decomposition (eDMD) [30]. DMD algorithms were initially used to decompose spatial-temporal coherent structures, enabling the solution or approximation of dynamical systems with structures that evolve, decay, and oscillate over time.

However, DMD relies on linear measurement functions and cannot adequately capture nonlinear changes in coordinates within strongly nonlinear systems. To address this limitation, Williams et al. [30] proposed extended dynamic mode decomposition (eDMD).

In eDMD, the traditional linear DMD regression is adapted not directly to raw data but to augmented matrices containing nonlinear measurement states.

Since the augmentation vector $\mathbf{z} \in \mathbb{R}^p$ may diverge significantly from the nonlinear measurement state X, Williams proposed a computational method that incorporates the kernel trick. This method draws inspiration from the kernel method used in DMD approximations to compute the Koopman operator, which has emerged as a pivotal research area [31–33]. Theoretically, the functions $\{\Theta_k\}_{k=1}^{p}$ form a robust basis for approximating Koopman operators. Under infinite data conditions, eDMD matrices converge to Koopman operators for subspaces consisting of mapping functions. However, if these functions do not span Koopman-invariant subspaces, the projection operators may yield spurious eigenvalues and eigenvectors that differ from those of the true operators [34]. Therefore, validation and cross-validation are essential to prevent overfitting. Additionally, methods like the Sparse Identification of Nonlinear Dynamics (SINDy) [35], HAVOK [16], and Hankel-DMD [36] offer similar approaches.

In summary, dynamic mode decomposition (DMD) serves as an initial straightforward algorithm. Extended DMD (eDMD) requires an additional input of basis functions, or dictionaries, within the Koopman operator's subspace. The Sparse Identification of Nonlinear Dynamics (SINDy) employs sparse identification for simpler system modeling, HAVOK adapts well to chaotic systems without a predefined basis, and Hankel-DMD is suited for turbulent and quasi-periodic iterative system problems. Recently, scholars have increasingly explored neural networks for developing Koopman prediction models [37,38] and have focused on anomaly detection algorithms based on Koopman operator theory [39]. Ishikawa et al. [40] proposed an estimation method called JetDMD (Jet Dynamic Mode Decomposition), which enhances the estimation of the Koopman operator using the intrinsic structure of Reproducing Kernel Hilbert Spaces (RKHSs) and geometric concepts related to jets. This method improves upon EDMD (empirical dynamic mode decomposition) particularly in numerical estimation of eigenvalues. While the incorporation of RKHS methods represents a novel approach, it currently only considers exponential and Gaussian kernels, leaving room for exploration in other directions.

In practice, modern techniques for representing dynamical systems prioritize identifying relevant eigenfunctions for prediction rather than merely approximating Koopman invariant subspaces [41]. While current data-driven methods effectively capture transient and quasi-periodic behaviors of nonlinear systems, further research is needed for nondissipative dynamical systems and continuous spectra. The efficient mastery of this computational paradigm could substantially simplify downstream analysis and control tasks. This motivates a new research direction: DeepKoopman, the deep neural Koopman operator [42,43]. This approach aims to develop lightweight, mesh-independent neural operator models using Koopman's global linearization theory combined with neural operators to solve the partial differential equations (PDEs) of nonlinear systems. Solving scientific PDEs is a fundamental challenge in many fields, and the neural operator method, based on the Koopman operator, combines theoretical completeness with the high performance of neural networks. Future developments in this area are anticipated.

2.2. Neural Operator

Neural operators are not a new concept, originating from physics-informed AI (PIM) research, where physics-informed neural network (PINN) [44] approaches are most prominent. The latest designs for solving partial differential equations (PDEs) or modeling dynamic systems are now beginning to incorporate Fourier [45,46] neural operators or Koopman neural operators, with the DeepONet method proposed by Lu in 2019 [47]. These methods not only learn the solution operators of the overall PDE family but also determine solutions to nonlinear dynamic systems of equations. The essence of their operator learning is to improve the neural network so that it learns not just an ordinary approximation function, but a mapping relation, i.e., an operator. The generalization error is significantly reduced compared to fully connected networks, which encompass a complete set of approximation modeling workflows for arbitrary dynamical systems. In other words, training a well-established network structure with neural operators using measurement data can result in generalized dynamical system models. Theoretically, trajectories generated by dynamical systems should also be modeled and predicted.

2.3. Trajectory Prediction

Trajectory data consist of a series of chronologically arranged points, sampled from the motion process of one or more moving objects. They typically include location information (latitude, longitude, altitude, etc.), timestamps, and other attributes such as velocity, acceleration, direction, and power parameters. Trajectories can be categorized based on predicted outputs as uni-modal, multi-modal, or interactive. According to motion characteristics, they can also be classified as linear or nonlinear trajectories. Nonlinear trajectories are sequences of continuous or discrete observable points generated by nonlinear dynamical systems over time, commonly observed in the real world. Flight trajectories, for instance, fall under this category. From a dynamics standpoint, vehicle motion descriptions require nonlinear dynamics equations incorporating mass and rigid body dynamics and that are influenced by various aerodynamic forces and moments. Modeling these equations is highly complex. Additionally, vehicle dynamic behavior is represented by state-space models encompassing variables such as position, velocity, acceleration, and attitude, necessitating the consideration of multiple parameters. Vehicle dynamical systems are highly nonlinear and involve uncertainties like model errors, sensor noise, and external perturbations, significantly complicating prediction tasks. Traditional physical models may not fully capture all dynamic behaviors. While machine learning and data-driven approaches can learn flight patterns and behaviors from historical data, effectively integrating these methods remains a challenge. Fortunately, more and more attention is now being paid to combining machine learning and other methods for flight trajectory prediction. Addressing the prediction uncertainties caused by various interferences has also become a hot research topic. Shafienya et al. [48] proposed a hybrid deep learning model combining a CNN-GRU and 3DCNN for 4D flight trajectory prediction. This model excels in extracting and predicting spatio-temporal features, offering high prediction accuracy, extended prediction time, and comprehensive spatio-temporal feature extraction. Jia et al. [49] introduced an attention-based LSTM trajectory prediction model that utilizes a sliding window approach, which enhances the model training efficiency. Choi et al. [50] presented a trajectory prediction framework integrating machine learning and physics. Their approach, called the Residual Mean Interaction Multiple Model (RIMM), combines machine learning predictions with measurements using a physics-based estimation algorithm to explain current aircraft motions.

Nonlinear systems involve independent variables in a unique form of change, resulting in a mapping relationship that differs from traditional ones. This may include iterative functions and algorithms where the mapping of independent variables from one step to the next cannot be directly expressed using a linear function. It can be said that a linear relationship is a mutually exclusive and independent connection, while a nonlinear relationship embodies interactions. The central mathematical feature here is that the superposition theorem no longer applies. It is defined as follows: If the operator $N(\varphi)$ does not satisfy the equation $\mathcal{L}(a\varphi + b\psi) = a\mathcal{L}(\varphi) + b\mathcal{L}(\psi)$ for some a, b or φ, ψ, then it is a nonlinear operator.

The study of multivariate nonlinear time series, stemming from nonlinear dynamical systems, has a long and challenging history. This is because it is often crucial to analyze complex nonlinear dynamical systems across various scientific fields, such as the trajectory of a double pendulum system, the pattern of geomagnetic variation, or the human electrocardiogram (as referenced in [16]). These dynamical systems reflect real-world phenomena influenced by multiple variables. Directly observing time series from only a subset of these variables typically does not allow for the direct establishment of a comprehensive analytical mathematical model. Therefore, the study of complex dynamical systems often focuses on analyzing these time series, leading to the emergence of research into nonlinear multivariate time series. However, there is currently no unified mathematical framework explicitly explaining and describing their general characteristics.

In summary, we leverage the measure invariance of the Koopman neural operator, combined with dynamical system identification methods, to develop and deploy a comprehensive prediction model with interpretability using a data-driven approach, without requiring additional human intervention.

3. Deep Embedding Koopman Neural Operator (DE-KNO)

3.1. General Framework

Following the convention, the general framework can be divided into three parts: the encoder, the approximator, and the decoder.

The Koopman operator is defined as follows in reference [42]:

$$\mathcal{K}^{\varepsilon}\Theta(g_t) = \Theta(g_{t+\varepsilon}), \forall g_t \in \mathbb{R}^{d_g} \times T \tag{4}$$

where g is the state variable of the dynamical system mapped to the space of observable functions, encoded from the sampling points by the encoder. The subscript t represents the moment, $\varepsilon \in \mathbb{Z}$ represents the number of evolutionary steps over time, T represents the overall trajectory time length, $\mathcal{K}$ is the infinite-dimensional linear Koopman operator, and Θ represents the observation function. The learning objective of this model is the observation function Θ and the Koopman operator $\mathcal{K}$, and the general framework is shown in Figure 3.

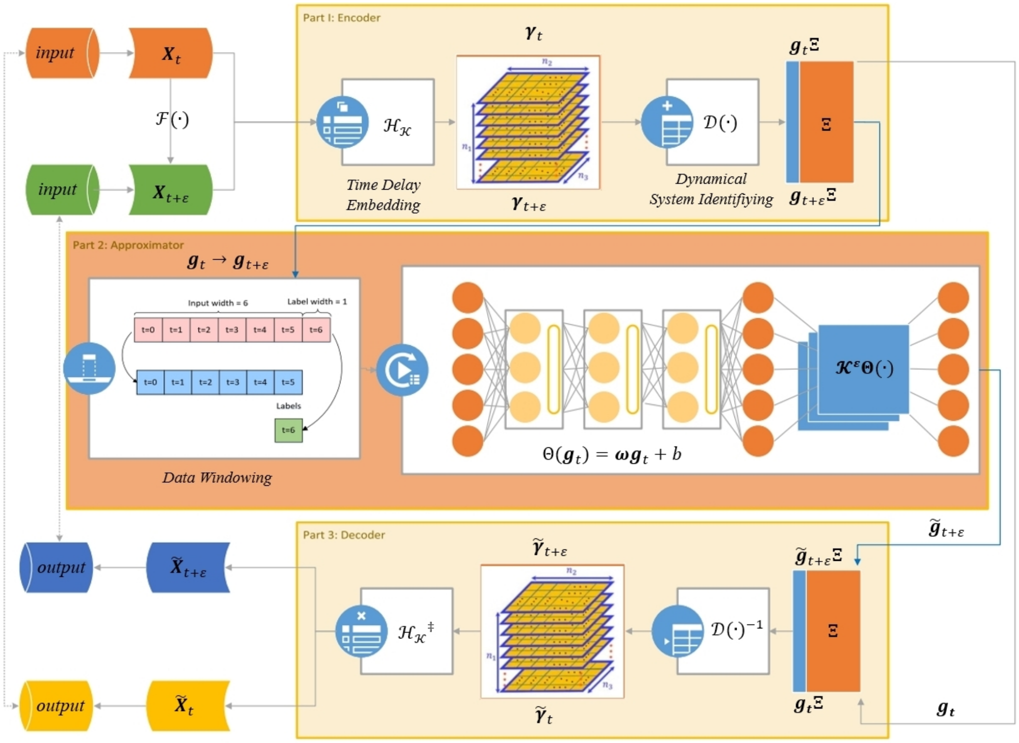


Figure 3. Deep Embedding Koopman Neural Operator (DE-KNO) general framework.

The whole-model forward prediction is

$$\tilde{X}_{t+\varepsilon} = \mathcal{H}_{\mathcal{K}}^{\ddagger}\left(\mathcal{D}^{-1}(K^{\varepsilon}\Theta(\mathcal{D}(\mathcal{H}_{\mathcal{K}}(X_t)))\Xi)\right) \tag{5}$$

The overall loss function is

$$\mathcal{L}oss_{all} = \alpha + \beta \tag{6}$$

$$\alpha = \sum_{t=1}^{T} \|\widetilde{X}_t - X_t\|^2 = \sum_{t=1}^{T} \|\mathcal{H}_{\mathcal{K}}{}^{\ddagger}\Big(\mathcal{D}^{-1}(\mathcal{D}(\mathcal{H}_{\mathcal{K}}(X_t)))\Big) - X_t\|^2 \tag{7}$$

$$\beta = \sum_{t=1}^{T} \|\mathcal{H}_{\mathcal{K}}{}^{\ddagger}\Big(\mathcal{D}^{-1}(K^{\varepsilon}\Theta(\mathcal{D}(\mathcal{H}_{\mathcal{K}}(X_t)))\Xi)\Big) - X_{t+\varepsilon}\|^2 \tag{8}$$

where X_t represents a moment in the original state $X \in \mathbb{R}^{T\times d}$, $X_{t+\varepsilon}$ represents the actual evolutionary outcome state, $\widetilde{X}_{t+\varepsilon}$ represents the evolutionary outcome predicted by the model, $H_{\mathcal{K}}$ is the Koopman modified form of the Hankel matrix for time-delayed embedding, and $\mathcal{H}_{\mathcal{K}}{}^{\ddagger}$ is the Hankel matrix restoring operation; α and β are the weighting values, where α is driven to zero using optimization to learn the approximate Koopman operator, and β is driven to zero using optimization to learn the dictionary function of the recognition system.

The encoder's task is to capture variables in the new function space and transform the input space into a measurement-invariant function space through the preprocessing of the input data, time-embedded coding, and identification of the dynamical system.

The role of the approximator is to approximate the Koopman operator that drives the system forward and the observation function, enabling the provision of physical explanations through the visualization of the operator module.

The decoder serves as the inverse of the encoder, restoring the function space to the original space. The summary of the DE-KNO model is presented in Algorithm 1.

Algorithm 1 DE-KNO

```
Input: Time series data $X_t$ and $X_{t+\varepsilon}$
Output: Predicted data $\widetilde{X}_{t+\varepsilon}$ and $\widetilde{X}_t$
Step 1: Encoder
Apply temporal delay embedding to input time series data $X_t$, generating embedding vector $\gamma_t$.
Convert embedding vector $\gamma_t$ into state variable $\gamma_t$ using the dynamical system identification module $D(\cdot)$.
Generate dictionary function space variables $g_t$ and its derivative $\dot{g}_t$ based on state variable $\gamma_t$.
Step 2: Approximator
Perform windowing on input data to generate training data and labels.
Train observation function $\Theta$, mapping dictionary function space variable $g_t$ to the observation space.
Compute the approximate matrix $K$ for the Koopman operator $\mathcal{K}$.
Alternate training of observation function $\Theta$ and the approximate matrix $K$ of the Koopman operator $\mathcal{K}$, optimizing the loss function $L$.
Step 3: Decoder
Predict the future state of the system using predicted observation space variable $\widetilde{g}_{t+\varepsilon}$ through observation function $\Theta$ and Koopman operator $\mathcal{K}$.
Reverse decode predicted observation space variable $\widetilde{g}_{t+\varepsilon}$ using the inverse of dynamical system identification module $D^{-1}(\cdot)$, restoring it to the original state variable $\widetilde{\gamma}_{t+\varepsilon}$.
Use the inverse matrix $\mathcal{H}_{\mathcal{K}}^{\ddagger}$ of the temporal delay embedding to retrieve original input space data $\widetilde{X}_{t+\varepsilon}$ from state variable $\widetilde{\gamma}_{t+\varepsilon}$.
Output: $\widetilde{X}_{t+\varepsilon}$ and $\widetilde{X}_t$.
```

3.2. Encoder

The practical implication of Koopman's theory is that we do not need to measure the actual state of the system. Instead, we seek functions from the eigenspace of the

Koopman operator as our measurements, thereby transforming the task of nonlinear system identification into one of linear system identification. However, acquiring the eigenspace of the Koopman operator or the invariant subspace of the state variable locus remains a challenge. While we can find them through direct observation for some simple nonlinear systems assuming the form of the dynamical equations is known, it is exceedingly difficult for most nonlinear systems. Even if discovered, they often result in an infinite-dimensional linear system.

According to Takens' embedding theorem, for an infinite-length, noise-free scalar time series of d-dimensional chaotic attractors $\{x(t)\}$, we can always find an n-dimensional embedding phase space in a topologically invariant sense as long as $n \geq 2d+1$. This theorem guarantees that we can reconstruct a phase space equivalent to that of its prime mover from a chaotic sequence, in a topological sense. Packard et al. [51] proposed both derivative reconstruction and coordinate delay reconstruction for the phase space reconstruction of chaotic nonlinear trajectories. However, in practical applications, a posteriori information is often lacking, and numerical differentiation is too error-sensitive. Therefore, most applications utilize the coordinate-delayed phase space reconstruction method.

Under certain assumptions, Takens' embedding theorem allows us to recover complete system information by observing a system state along with a delayed embedding method. This method is considered applicable in other nonlinear systems. Hence, we employ a trick: construct the Hankel matrix to approximate the full state of the dynamical system using the time delay embedding method associated with the coordinate-delayed phase space reconstruction technique.

The inputs to this model are the following nonlinear trajectories:

$$X_t = \left[x_t^{(1)} x_t^{(2)} \cdots x_t^{(d)}\right]_{1\times d} \tag{9}$$

$$X_{t+\varepsilon} = \left[x_{t+\varepsilon}^{(1)}\ x_{t+\varepsilon}^{(2)} \cdots\ x_{t+\varepsilon}^{(d)}\right]_{1\times d} \tag{10}$$

The Hankel matrix is generated after time-delayed embedding, where the embedding dimensions are $n = (\tau+1)d$, $\tau \geq 1$:

$$H_{\mathcal{K}}(X_t) = \begin{bmatrix} X_t & X_{t+1} & \cdots & X_{t+\tau} \\ X_{t+1} & X_{t+2} & \cdots & X_{t+\tau+1} \\ \vdots & \vdots & \vdots & \vdots \\ X_{t+\tau} & X_{t+\tau+1} & \cdots & X_{t+2\tau} \end{bmatrix}_{(\tau+1)d\times(\tau+1)d} = \gamma_t \tag{11}$$

$$H_{\mathcal{K}}(X_{t+\varepsilon}) = \begin{bmatrix} X_{t+\varepsilon} & X_{t+\varepsilon+1} & \cdots & X_{t+\varepsilon+\tau} \\ X_{t+\varepsilon+1} & X_{t+\varepsilon+2} & \cdots & X_{t+\varepsilon+\tau+1} \\ \vdots & \vdots & \vdots & \vdots \\ X_{t+\varepsilon+\tau} & X_{t+\varepsilon+\tau+1} & \cdots & X_{t+\varepsilon+2\tau} \end{bmatrix}_{(\tau+1)d\times(\tau+1)d} = \gamma_{t+\varepsilon} \tag{12}$$

The final generated tensor full state $\gamma \in \mathbb{R}^{\mathbf{T}\times(\tau+1)d\times(\tau+1)d}$ can represent the overall physical state of the dynamical system, where $\mathcal{T} = T - d + 2$. However, there is still no clear physical explanation, and it is still a nonlinear system, so we analogize the SINDy algorithm idea through the selection of data-driven dictionary libraries without the need for physical a priori knowledge (of course, CustomLibrary can be added) and according to the dynamical system identification method to learn the appropriate augmentation and generalization functions through sparse regression methods to obtain the full state of the system model.

We utilized PolynomialLibrary and FourierLibrary as a foundation for deriving dictionary functions and generalization functions following dynamical system identification:

$$g_t = \begin{bmatrix} | & | & | & | & | & | & | & | & | \\ 1 & \gamma_t & {\gamma_t}^2 & \cdots & sin(\gamma_t) & cos(\gamma_t) & sin(2\gamma_t) & cos(2\gamma_t) & \cdots \\ | & | & | & | & | & | & | & | & | \end{bmatrix} \tag{13}$$

$$\dot{\gamma}_{d\times 1} = \mathcal{D}(\gamma) = \Xi_{d\times N} g_{N\times 1} \tag{14}$$

$$\dot{\gamma} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_d \end{bmatrix} = \begin{bmatrix} \Xi_{0,0} & \cdots & \Xi_{0,N-1} \\ \vdots & \ddots & \vdots \\ \Xi_{d-1,0} & \cdots & \Xi_{d-1,N-1} \end{bmatrix}_{d\times N} \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_d \\ {\gamma_0}^2 \\ \vdots \\ sin(\gamma_0) \\ cos(\gamma_0) \\ \vdots \\ cos(m\gamma_d) \end{bmatrix}_N = \Xi_{d\times N} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_N \end{bmatrix} \tag{15}$$

The state variables of the dynamical system at this time have been mapped from the nonlinear γ_t to the g_t of the dictionary function space, and the variables of this function space have eliminated the nonlinear characteristics to a certain extent, where $\mathcal{D}$ is the augmentation and generalization function, Ξ is the matrix of the discrimination parameter, and N is the total number of the variables of g_t of the dictionary function space, and we define the dynamical system at this time to be

$$\mathcal{S}_1 : (\mathcal{P}, \mathcal{T}, \mathcal{D}), \mathcal{P} \subseteq \mathbb{R}^N, \mathcal{T} \in \mathbb{Z}, \mathcal{D} : \mathcal{P} \to \mathcal{P} \tag{16}$$

where $\mathcal{P}$ is the phase space, and the values therein are the states. $\mathcal{T} \in \mathbb{Z}$ is discrete time, $\mathcal{D}$: $\mathcal{M} \to \mathcal{M}$ is an evolution operator, and F$\gamma_t = \gamma_{t+1}$, where $\gamma_t, \gamma_{t+1} \in \mathcal{M}$. $\gamma_t \in \mathcal{M}$ represents a flight state using the evolution operator F.

3.3. Approximator

The Koopman operator is a mapping of functions from the state space to functions in the state space, rather than from states to states. It essentially defines a completely new dynamical system:

$$\mathcal{S}_2 : (\mathcal{Q}, \mathcal{T}, \mathcal{K}), \mathcal{Q} \subseteq \mathbb{C}^{N_k}, \mathcal{T} \in \mathbb{Z}, \mathcal{K} : \mathcal{Q} \to \mathcal{Q}, \Theta \in \mathcal{Q}, \Theta : \mathcal{P} \to \mathcal{Q} \tag{17}$$

controls the evolution of the observables in discrete time. Where $\mathcal{Q}$ is the space of observable functions, unlike the $\mathcal{P}$ state space, the operator $\mathcal{K}$ acts on the function, and Θ is the observable function responsible for the spatial mapping of the $\mathcal{P} \to \mathcal{Q}$ so that even though Θ is finite-dimensional nonlinear, $\mathcal{K}$ is still infinite-dimensional linear.

Fortunately, the finite-dimensional matrix K of the approximation approximation is reasonably truncated $\mathcal{K}$, which does not lose much accuracy. Therefore, the actual computation to find its approximation matrix $K \in \mathbb{C}^{N\times N}$ for approximating the $\mathcal{K}$ operator is conducted by taking the encoder-generated dictionary function space variable g_t as the input to the approximator and feeding the training data to the two parts of the approximator, the regular block and the operator block, in batches using the sliding window method. The regular block can be customized with any neural network layer to be superimposed for finding the observation function Θ in the space of g_t, and the operator block is used to compute the approximation matrix K. The two blocks are trained alternately using an iterative method, and finally, the trained neural network is obtained as the observation function

Θ, which is used jointly with the approximation matrix K to realize the reconstruction of the dynamical system.

For the dynamical system $\mathcal{S}_1$, where phase space $\mathcal{M}$ is a state space, we consider this dynamical system of the form

$$\frac{d}{dg}g(t) = D(g(t)) \tag{18}$$

Its induced discrete dynamical systems are

$$g_{t+1} = \mathcal{D}(g_t) \tag{19}$$

We know that according to Koopman operator theory,

$$\mathcal{K}\Theta \triangleq \Theta \circ \mathcal{D} \Rightarrow \mathcal{K}\Theta(g_t) = \Theta(g_{t+1}) \tag{20}$$

$$g_{t+1} = \mathcal{D}(g_t) = \sum_{k=1}^{N_k} \nu_k \varphi_k(g_t) \tag{21}$$

$$\mathcal{D}(g) = \Theta(g)a \tag{22}$$

where ν_k is the Koopman mode, and φ_k is the kth Koopman eigenfunction, $D \in \mathcal{D}$. With (3), it is known that the evolutionary values can be obtained through operator advancement:

$$\widetilde{g}_{t+\varepsilon} = \Theta(\widetilde{g}_{t+\varepsilon}) = \mathcal{K}^{\varepsilon}\Theta(g_t) = (\Theta \circ \mathcal{D})(g_t) = \Theta(g_t)(K^{\varepsilon}a) + r(g_t) \tag{23}$$

To determine K, we need to minimize the residuals:

$$\mathcal{J} = \|((\Theta \circ \mathcal{D})(g_t) - \Theta(g_t)K^{\varepsilon})a\|^2 = \|(\Theta(g_{t+\varepsilon}) - \Theta(g_t)K^{\varepsilon})a\| \tag{24}$$

Since both K and Θ are unknown, the overall training object is

$$(\mathcal{K}, \Theta) = argmin\mathcal{J}\left(K, \widetilde{\Theta}\right) = \sum_{\varepsilon=1}^{\mathcal{T}} \|\widetilde{\Theta}(g_{t+\varepsilon}) - \widetilde{\Theta}(g_t)K^{\varepsilon}\|^2 + \lambda\|K\|^2 \tag{25}$$

where the loss function is

$$\mathcal{L}oss = \sum_{\varepsilon=1}^{\mathcal{T}} \|\widetilde{\Theta}(g_{t+\varepsilon}) - \widetilde{\Theta}(\widetilde{g}_{t+\varepsilon})\|^2 = \sum_{\varepsilon=1}^{\mathcal{T}} \|\widetilde{\Theta}(g_{t+\varepsilon}) - \widetilde{\Theta}(g_t)K^{\varepsilon}\|^2 \tag{26}$$

In referring to the EDMD algorithm, the least squares method is used to minimize Equation (21), where the Koopman operator $\mathcal{K}$ can be approximated as K:

$$K \triangleq \mathcal{G}^{+}\mathcal{A} \tag{27}$$

where $\mathcal{G}^{+}$ is pseudo-reverse

$$\mathcal{G} = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \widetilde{\Theta}(g_t)^{T} \cdot \widetilde{\Theta}(g_t) \tag{28}$$

$$\mathcal{A} = \frac{1}{\mathcal{T}} \sum_{t=0,\varepsilon=1}^{\mathcal{T}} \widetilde{\Theta}(g_t)^{T} \cdot \widetilde{\Theta}(g_{t+\varepsilon}) \tag{29}$$

We use a fully connected deep neural network (DNN) with three hidden layers to reconstruct a new variable space, which is common for regression tasks. The observation function Θ is then obtained by parameterizing the dense neural network:

$$\Theta = \omega^{(out)}h^{(3)} + b^{(out)} \tag{30}$$

$$h^{l+1} = tanh\left(\omega^{(l)}h^{(l)} + b^{(l)}\right), l = 0, 1, 2 \tag{31}$$

where $\omega^{(out)}$ is the weight of the output layer, $b^{(out)}$ is the bias of the output layer, $h^{(3)}$ is the value of the third layer, and h^{l+1} represents the value of the $l+1$ layer obtained through an activation function from the lth layer. Θ is important as the hidden projection function in DNN regression and as the spatial mapping function connecting the reconstructor and predictor. Through training, the observation function Θ and the approximation matrix K are successively obtained, and the forward prediction is obtained:

$$\widetilde{g}_{t+\varepsilon} = \widetilde{\Theta}(g_{t+\varepsilon}) = \widetilde{\Theta}(g_t)K^{\varepsilon} \tag{32}$$

3.4. Decoder

The decoder is the inverse of the encoder, restoring the predicted $\widetilde{g}_{t+\varepsilon}$ back to the original space:

$$\widetilde{\gamma}_{t+\varepsilon} = \mathcal{D}^{-1}(\widetilde{g}_{t+\varepsilon}) \tag{33}$$

$$\widetilde{X}_{t+\varepsilon} = \mathcal{H}_{\mathcal{K}}^{\ddagger}(\widetilde{\gamma}_{t+\varepsilon}) \tag{34}$$

where $\mathcal{D}^{-1}(\cdot)$ is the inverse of the generalized function, and $\mathcal{H}_{\mathcal{K}}^{\ddagger}(\cdot)$ denotes Hankel matrix restoration operations.

As such, the data complete the end-to-end prediction process from Hilbert space to phase space to observer observation space, and then from observation space back to phase space and finally back to Hilbert space.

4. Experimental Studies

4.1. Preparation

For the datasets, we utilized two typical nonlinear physical systems, namely Lorenz and Rossler, along with seven real-world benchmark datasets: ETT [52], Exchange [53], four datasets provided in [54] (Electricity, ILI (CDC), Traffic, Weather (https://drive.google.com/drive/folders/1ZOYpTUa82_jCcxIdTmyr0LXQfvaM9vIy, accessed on 1 December 2023)), ECG (https://www.heywhale.com/mw/dataset/5d678efa8499bc002c08c8f4/file, accessed on 1 December 2023), EEG (https://www.kaggle.com/datasets/samnikolas/eeg-dataset, accessed on 1 December 2023), and the CAFUC dataset [13,55] generated from our collection, as shown in Figure 4. The CAFUC dataset consists of data generated by flight students during their flight learning and training. The data have 64 dimensions. In this experiment, we used data generated by students during rectangular takeoff and landing exercises using the same model of aircraft. Unlike common 4D data, we have more dimensions, but in this experiment, we used the following dimensions: Latitude, Longitude, AltMSL (altitude above mean sea level), Pitch, Roll, LatAc (lateral acceleration variation), NormAc (normal acceleration variation), HDG (magnetic heading), TRK (track angle), E1RPM (engine 1 revolutios per minute), E1CHT1 (engine 1 cylinder head temperature 1), and E1EGT1 (engine 1 exhaust gas temperature 1). In the dataloader section, data extraction was performed on the input data to obtain the training set and the test set, with the test set being 20 percent of the entire dataset. This created a total of 11 datasets for comparison experiments, with reference to [16,56].

For the baseline, we selected representatives from five classes of models: Koopman-based KNF and Koopa, MLP-based Dlinear and LSTM, Fourier-based FiLM, TCN-based TimesNet, and Transformer-based Autoformer, totaling seven models for comparison in our prediction experiment. All experiments were conducted using the publicly available source code provided by the original authors. The models are introduced as follows:

(1) KNF (Koopman Neural Forecaster) [57] is a novel method rooted in Koopman theory adept at accurately predicting time series even amidst distribution changes. It harnesses the Koopman matrix to capture global behaviors that evolve over time, adapting to local distribution shifts.

(2) Koopa [56] consists of modular Koopman Predictors (KPs) designed to hierarchically describe and propel the dynamics of time series. It employs Fourier analysis for

disentangling dynamics. For time-invariant behaviors, the model learns Koopman embedding and linear operators to uncover implicit transitions underpinning long-term series.

(3) The FiLM (Frequency Improved Legendre Memory) [58] architecture integrates a mixture of experts, tailored for robust multiscale feature extraction from time series. It reconfigures the Legendre Projection Unit (LPU), making it a versatile tool for data representation. This adaptation allows any time series forecasting model to utilize the LPU effectively while preserving historical information.

(4) Timesblock, part of timesnet [59], adaptively transforms 1D time series into a set of 2D tensors based on learned periods. It further captures intra-period and inter-period variations in the 2D space using a parameter-efficient inception block.

(5) Autoformer [54] preserves the residual and encoder–decoder structure of Transformers but innovates with a decomposition forecasting architecture. It embeds proposed decomposition blocks as inner operators to progressively isolate long-term trend information from predicted hidden variables. Through replacing self-attention with an auto-correlation mechanism, Autoformer identifies sub-series similarities based on periodicity and aggregates similar sub-series from previous periods.

The experimental equipment included an RTX-A4000 graphics card. Furthermore, two of the most commonly used normalization functions in the literature were selected to preprocess the data: the min-max scaler and mean normalization, also known as the z-score. The complete grid of training parameters included a batch size of 64, two epochs, and the optimizers SSR and Adam, with a learning rate of 0.001.

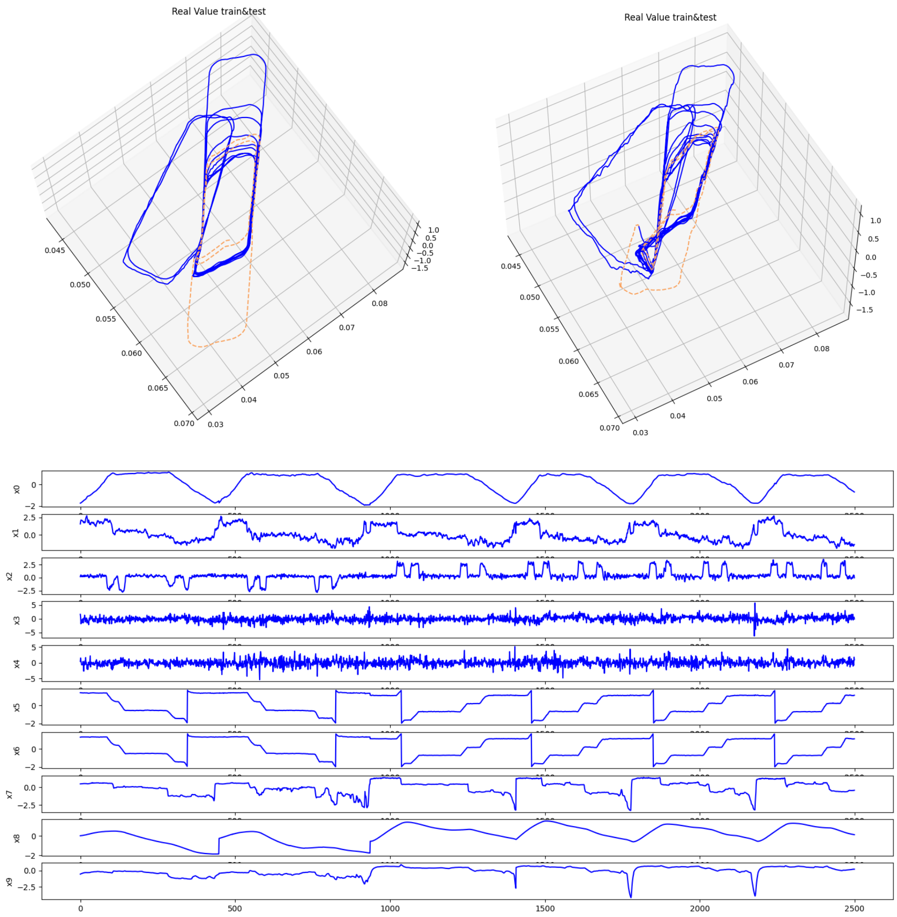


Figure 4. The CAFUC dataset is visualized with 3D trajectory views from different viewpoints, the blue line is the original part and the yellow line is the predicted part. (**top**) and 2D views showing the time series of the main parameters (**bottom**).

4.2. Process

In taking the CAFUC dataset as an example, the dataset is cleaned and normalized. Time-delayed embedding is performed, and the generalization function, variables, and their coefficients are obtained through dynamical system identification. The sliding window is then fed into the iterative training process to obtain the observation function and neural operator alternately, which is restored back to the original space after the inverse operation. In comparing this process to the classical physical Lorenz system, of which its main experimental process is visualized in Figure 5, it can be seen that both the real data and the physical system can be better predicted.

In terms of the detailed process, let us take the experiment process (a) for the CAFUC dataset as an example. (a.1) represents the attractor trajectory, (a.2) is the parameter time series, and (a.3) shows the violin plot for each parameter, illustrating the overall distribution of the multimodal data. (a.4) displays the violin plot for each parameter after the dictionary augmentation. (a.5) represents the prediction for a time window, (a.6) visualizes the Koopman operator approximation matrix K, (a.7) shows the predicted violin plots of each parameter, (a.8) displays the violin plots of each parameter after dictionary augmentation reduction, (a.9) compares the reduced predicted parameter tensors with the true values, and (a.10) illustrates the restored predicted attractor trajectories versus the true values. (b.1)–(b.10) is the same as (a.1)–(a.10).

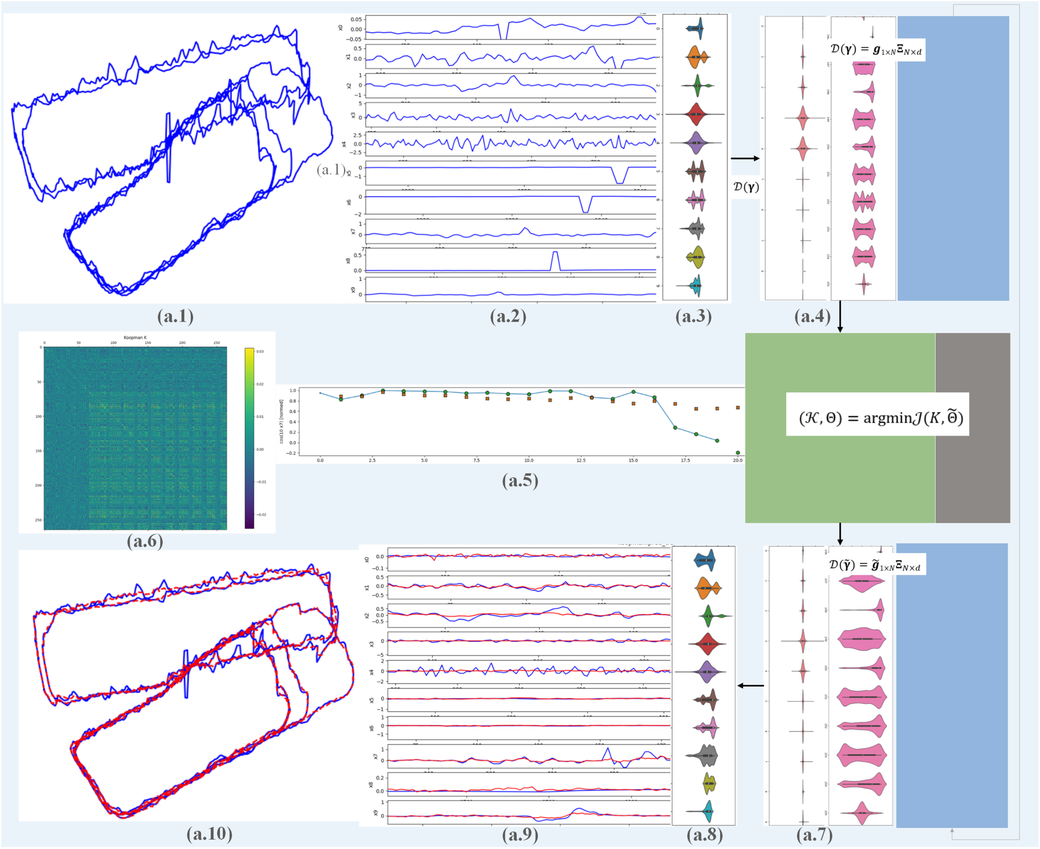


Figure 5. *Cont.*

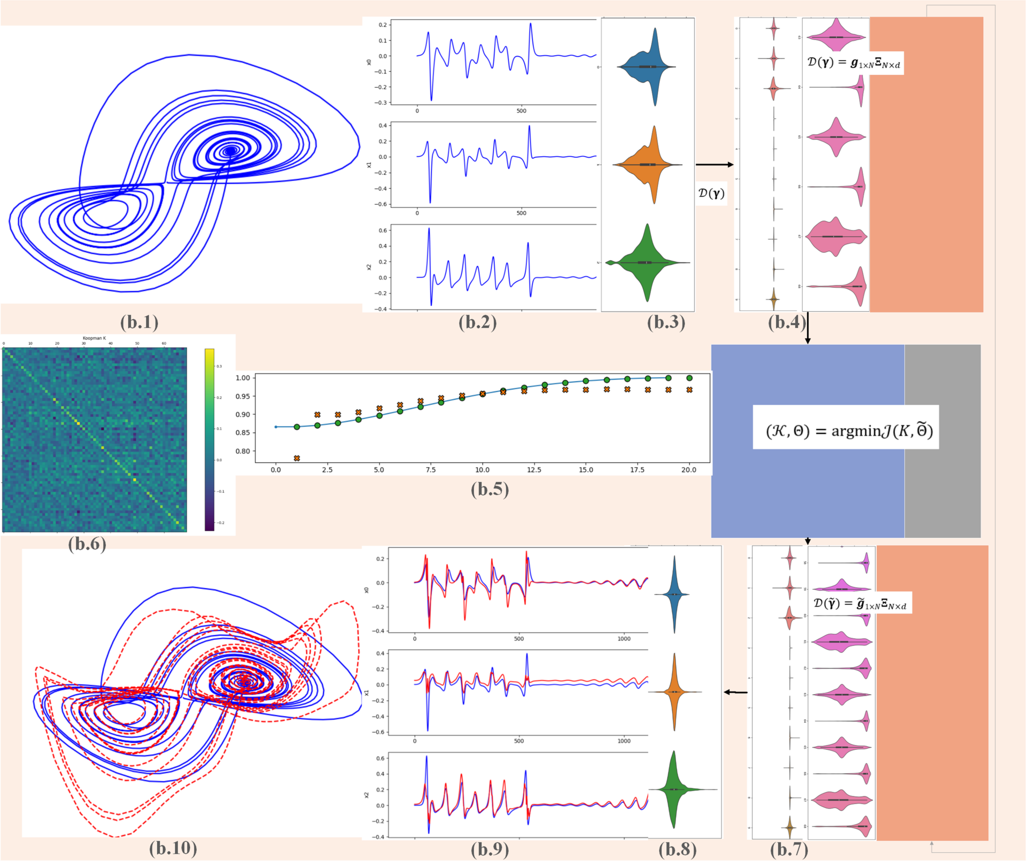

(b.1)
(b.2)
(b.3)
(b.4)
$\mathcal{D}(\gamma) = g_{1\times N}\Xi_{N\times d}$
$\mathcal{D}(\gamma)$
Koopman K
(b.6)
(b.5)
$(\mathcal{K},\Theta) = \text{argmin}\mathcal{J}(K,\widetilde{\Theta})$
$\mathcal{D}(\widetilde{\gamma}) = \widetilde{g}_{1\times N}\Xi_{N\times d}$
(b.10)
(b.9)
(b.8)
(b.7)

Figure 5. Experiment process: (**a**) CAFUC, (**b**) Lorenz.

Furthermore, we examined the correlation between the increase in data length and the corresponding change in model performance. The experimental results are depicted in Figure 6. According to the findings, the model's training time increases linearly with the dataset's length, suggesting that the model exhibits good stability.

To conduct a more comprehensive comparison between the proposed model and the benchmark model, we used a five-fold cross-validation repeated 10 times to obtain the MSEs of the models on the CAFUC dataset for a paired *t*-test to determine if there is a significant difference in their performances. We set the *p*-value to 0.05. The experimental results show that there is a significant difference between our model and the benchmark model.

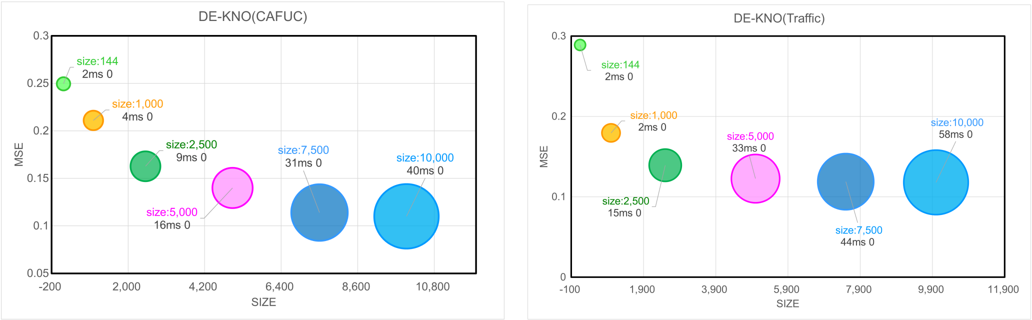


Figure 6. Model stability test results: **left**, efficiency on CAFUC dataset; **right**, efficiency on Traffic dataset.

4.3. Results

We conducted comparative experiments on our DE-KNO model with seven models known for their performance on temporal nonlinear datasets. Our evaluations included CA-

FUC's flight trajectory dataset, along with two classic nonlinear physical systems (Lorenz and Rossler) and six commonly used time series datasets such as brain wave EEG and heart rate ECG. These datasets were also utilized for training and testing with the benchmark models. Table 1 presents the experimental results, highlighting the best-performing models with bold formatting. Each model was evaluated on the same datasets to ensure a fair comparison. Detailed results for the primary datasets are shown in Table 2. In terms of efficiency on the CAFUC dataset, DE-KNO does not excel in training time but stands out with the second lowest MSE and the lowest MAE. While the MSE is not optimal, our model requires minimal memory resources for computation, and there is still room for improvement. DE-KNO also demonstrates excellent MSE and MAE performances on other datasets.

Table 1. The multivariate prediction results.

Model	Koopa		KNF		Dlinear		LSTM		FiLM		TimesNet		Autoformer		DE-KNO	
DataSets	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
CAFUC	**0.098**	0.194	0.589	0.874	0.736	0.680	0.350	0.403	0.851	0.640	0.197	0.284	0.592	0.597	0.117	**0.174**
Lorenz	0.187	0.270	0.935	0.917	0.873	0.749	0.126	0.223	1.152	0.846	0.491	0.681	1.006	0.789	**0.008**	**0.059**
Rossler	0.353	0.173	1.480	1.322	1.801	0.636	0.007	0.039	2.303	0.784	0.017	0.129	1.916	0.747	**0.003**	**0.003**
ETT	0.156	0.254	0.533	0.061	0.396	0.430	0.155	0.256	0.398	0.489	0.366	0.479	0.525	0.479	**0.148**	**0.246**
Electricity	**0.191**	**0.246**	0.936	1.227	0.224	0.316	0.879	0.649	0.253	0.272	1.755	1.363	0.263	0.314	0.287	0.366
Exchange	0.196	0.317	0.689	0.599	0.182	0.315	0.109	0.235	0.278	0.397	0.382	0.478	0.669	0.612	**0.010**	**0.046**
ILI	1.797	0.887	2.969	1.339	2.563	1.197	1.060	0.0.875	4.089	1.451	2.167	1.047	2.462	1.082	**0.369**	**0.481**
Traffic	0.464	0.296	0.687	0.437	0.459	0.371	0.439	0.390	0.467	0.291	0.647	0.392	0.710	0.397	**0.168**	**0.259**
Weather	0.198	0.269	0.483	0.479	0.240	0.279	0.049	0.148	0.233	0.274	0.291	0.297	0.336	0.383	**0.006**	**0.032**
ECG	0.750	0.510	3.156	2.397	0.950	0.550	0.871	0.499	0.854	0.560	1.914	1.148	0.807	0.533	**0.717**	**0.497**
EGG	0.507	0.519	2.234	2.505	0.528	0.520	0.925	0.621	0.591	0.561	2.033	1.676	0.663	0.602	**0.327**	**0.444**

The model's efficiency is primarily analyzed and compared in terms of training time and memory usage, including graphic memory usage. The comparison results are presented in Figure 7. It can be observed from the comparison in Figure 7 that our DE-KNO model exhibits the smallest error and a very low memory footprint, indicating a strong performance.

From Table 2, it can be observed that the K approximation matrices are completely different across different datasets. This indicates that the model has a certain degree of physical interpretability.

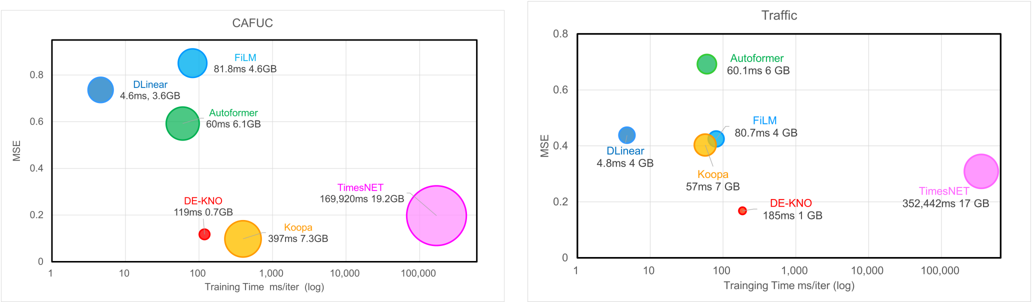
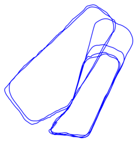
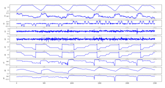
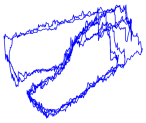
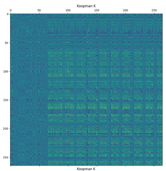
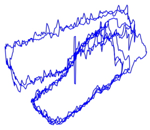
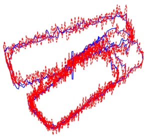
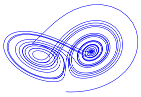
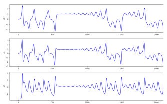
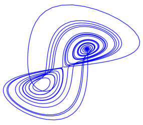
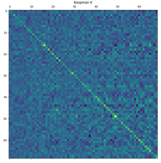
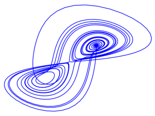
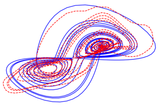
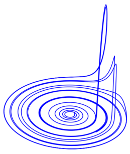
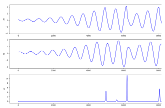
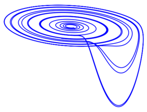
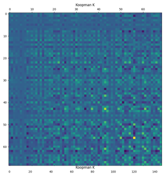
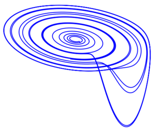
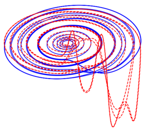
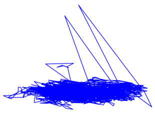
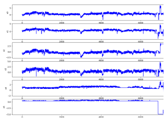
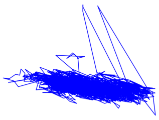
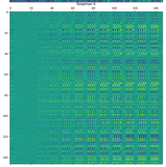
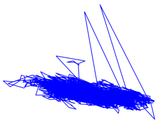
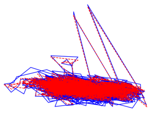
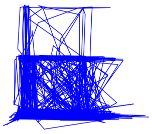
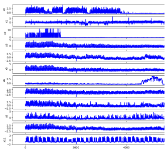
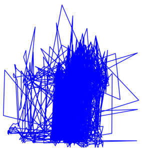
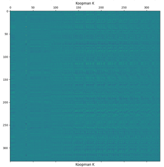
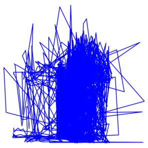
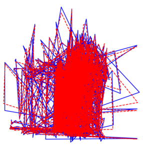
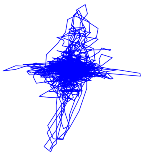
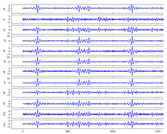
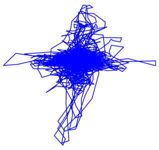
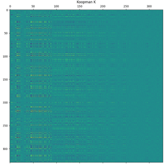
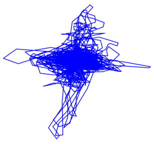
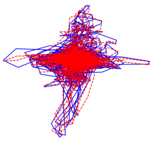


Figure 7. Model efficiency comparison: left, efficiency on CAFUC dataset; right, efficiency on Traffic dataset. The left graph records the training time, MSE, and memory usage of each model on the CAFUC dataset. The right graph records the training time, MSE, and memory usage of each model on the Traffic dataset.

Table 2. Some visualization results. The first column shows visualizations of various datasets, with the first row depicting rectangular takeoff and landing trajectories from the CAFUC dataset. The second column displays trajectories after the preprocessing of the raw data, while the third column shows embedded trajectories. The fourth column visualizes the approximation matrix K of the Koopman operator. The fifth and sixth rows depict reconstructed and predicted trajectories and The blue line is the original part and the red line is the predicted part.

Dataset	Time Series	Embedded Attractor	Koopman K	Reconstructed Attractor	Prediction Attractor
					
					
					
					
					
					

5. Conclusions

In this study, a data-driven modern Koopman operator theory and dynamical system identification method were employed to address the nonlinear flight trajectory prediction problem. The approach involves constructing a unified deep learning framework of Koopman neural operators with generalization ability, enabling end-to-end adaptive predictions using an operator push. Additionally, a certain level of physical interpretability is achieved through operator visualization and generative dictionary functions, which can be utilized for downstream pattern recognition and anomaly detection tasks. Compared to today's state-of-the-art methods, the model demonstrates good performance on several publicly available datasets, particularly on the flight trajectory dataset.

The datasets presented by the model were all derived from real training flight data. However, due to normalization applied during the data preprocessing stage, there is a risk of data distortion and pattern bias. This inadequacy in handling anomalies in actual data variations hinders accurate trajectory predictions. From the experimental observations, it is evident that the normalization technique we employed has significant flaws, which will be a focal point of our future research.

Our proposed model demonstrates promising theoretical prediction capabilities. However, the presence of pattern bias results in significant drawbacks during practical applications. We are committed to researching methods to eliminate pattern errors and enhance the model's accuracy in predicting data anomalies.

In the future, we will utilize Koopman neural operators for performing flight pattern recognition tasks, implementing online enhanced anomaly detection through pre-training for flight safety anomaly warning tasks and investigating the application of neural operators in real-world scenarios. As for the application, our plan involves integrating the model into PCB circuit boards and mounting them on training aircraft. This setup will enable the real-time reception of data during flight operations, facilitating trajectory prediction and the recognition of various training scenarios based on the acquired flight data. This provides a new approach for flight trajectory prediction and FOQA.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; validation, J.L.; formal analysis, J.L.; resources, J.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing review and editing, J.L.; visualization, J.J.; supervision, J.L.; project administration, Y.B.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China under grant 62272066; Sichuan Science and Technology Program under grants 2021JDJQ0021, 2022YFG0186, 2022NSFSC0511, and 2023YFG0027; Planning Foundation for Humanities and Social Sciences of Ministry of Education of China under grant 22YJAZH088; Chengdu Technological Innovation Research and Development Major Project under grant 2024-YF08-00029-GX; and Institute of Technology and Standards for Intelligent Management of Air Traffic Safety Project under JG2022-18.

Data Availability Statement: This research project was approved by the relevant ethical committee or institution and was conducted in strict accordance with ethical guidelines. In this study, we respected and protected the rights and privacy of the participants and ensured the confidentiality of their personal information.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zheng, Y. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 1–41. [CrossRef]
2. Georgiou, H.; Karagiorgou, S.; Kontoulis, Y.; Pelekis, N.; Petrou, P.; Scarlatti, D.; Theodoridis, Y. Moving objects analytics: Survey on future location & trajectory prediction methods. *arXiv* **2018**, arXiv:1807.04639.
3. Gavrilovski, A.; Jimenez, H.; Mavris, D.N.; Rao, A.H.; Shin, S.; Hwang, I.; Marais, K. Challenges and Opportunities in Flight Data Mining: A Review of the State of the Art. In Proceedings of the AIAA Infotech@ Aerospace, San Diego, CA, USA, 4–8 January 2016. [CrossRef]
4. Budalakoti, S.; Srivastava, A.; Otey, M. Anomaly Detection and Diagnosis Algorithms for Discrete Symbol Sequences with Applications to Airline Safety. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2009**, *39*, 101–113. [CrossRef]
5. Zheng, Y.; Zhou, X. (Eds.) *Computing with Spatial Trajectories*; Springer: New York, NY, USA, 2011. [CrossRef]
6. Huang, Y.; Du, J.; Yang, Z.; Zhou, Z.; Zhang, L.; Chen, H. A Survey on Trajectory-Prediction Methods for Autonomous Driving. *IEEE Trans. Intell. Veh.* **2022**, *7*, 652–674. [CrossRef]
7. Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep Learning for Time Series Classification: A Review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [CrossRef]
8. Zeng, W.; Chu, X.; Xu, Z.; Liu, Y.; Quan, Z. Aircraft 4D Trajectory Prediction in Civil Aviation: A Review. *Aerospace* **2022**, *9*, 91. [CrossRef]
9. Tang, J. Conflict Detection and Resolution for Civil Aviation: A Literature Survey. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 20–35. [CrossRef]
10. Ayhan, S.; Samet, H. Aircraft Trajectory Prediction Made Easy with Predictive Analytics. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 21–30. [CrossRef]
11. Zhang, X.; Mahadevan, S. Bayesian Neural Networks for Flight Trajectory Prediction and Safety Assessment. *Decis. Support Syst.* **2020**, *131*, 113246. [CrossRef]
12. Zhang, Z.; Guo, D.; Zhou, S.; Zhang, J.; Lin, Y. Flight Trajectory Prediction Enabled by Time-Frequency Wavelet Transform. *Nat. Commun.* **2023**, *14*, 5258. [CrossRef]

13. Lu, J.; Pan, L.; Deng, J.; Chai, H.; Ren, Z.; Shi, Y. Deep Learning for Flight Maneuver Recognition: A Survey. *Electron. Res. Arch.* **2023**, *31*, 75–102. [CrossRef]
14. Boril, J.; Jirgl, M.; Jalovecky, R. Using Aviation Simulation Technologies for Pilot Modelling and Flight Training Assessment. *Adv. Mil. Technol.* **2017**, *12*, 147–161. [CrossRef]
15. Zhang, J.; Zhang, P. *Time Series Analysis Methods and Applications for Flight Data*; Springer: Berlin/Heidelberg, Germany, 2017. [CrossRef]
16. Brunton, S.L.; Brunton, B.W.; Proctor, J.L.; Kaiser, E.; Kutz, J.N. Chaos as an Intermittently Forced Linear System. *Nat. Commun.* **2017**, *8*, 19. [CrossRef] [PubMed]
17. Rudenko, A.; Palmieri, L.; Herman, M.; Kitani, K.M.; Gavrila, D.M.; Arras, K.O. Human Motion Trajectory Prediction: A Survey. *Int. J. Robot. Res.* **2020**, *39*, 895–935. [CrossRef]
18. Schwarting, W.; Alonso-Mora, J.; Paull, L.; Karaman, S.; Rus, D. Safe Nonlinear Trajectory Generation for Parallel Autonomy with a Dynamic Vehicle Model. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2994–3008. [CrossRef]
19. Kobilarov, M. Nonlinear Trajectory Control of Multi-body Aerial Manipulators. *J. Intell. Robot. Syst.* **2014**, *73*, 679–692. [CrossRef]
20. Li, D.; Du, L. AUV Trajectory Tracking Models and Control Strategies: A Review. *J. Mar. Sci. Eng.* **2021**, *9*, 1020. [CrossRef]
21. Allgöwer, F.; Findeisen, R.; Nagy, Z. Nonlinear Model Predictive Control: From Theory to Application. *J. Chin. Inst. Chem. Eng.* **2004**, *35*, 299–315.
22. Koopman, B.O. Hamiltonian Systems and Transformation in Hilbert Space. *Proc. Natl. Acad. Sci. USA* **1931**, *17*, 315–318. [CrossRef] [PubMed]
23. Koopman, B.O.; Neumann, J. Dynamical Systems of Continuous Spectra. *Proc. Natl. Acad. Sci. USA* **1932**, *18*, 255–263. [CrossRef]
24. Neumann, J.v. Physical Applications of the Ergodic Hypothesis. *Proc. Natl. Acad. Sci. USA* **1932**, *18*, 263–266. [CrossRef]
25. Neumann, J.V. Proof of the quasi-ergodic hypothesis. *Proc. Natl. Acad. Sci. USA* **1932**, *18*, 70–82. [CrossRef] [PubMed]
26. Lange, H.; Brunton, S.L.; Kutz, N. From Fourier to Koopman: Spectral Methods for Long-term Time Series Prediction. *arXiv* **2020**, arXiv:2004.00574. https://doi.org/10.48550/arXiv.2004.00574.
27. Daneshfar, F.; Soleymanbaigi, S.; Nafisi, A.; Yamini, P. Elastic deep autoencoder for text embedding clustering by an improved graph regularization. *Expert Syst. Appl.* **2024**, *238*, 121780. [CrossRef]
28. Nasiri, E.; Berahmand, K.; Rostami, M.; Dabiri, M. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Comput. Biol. Med.* **2021**, *137*, 104772. [CrossRef] [PubMed]
29. Schmid, P.J. Dynamic Mode Decomposition of Numerical and Experimental Data. *J. Fluid Mech.* **2010**, *656*, 5–28. [CrossRef]
30. 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. [CrossRef]
31. Klus, S.; Nüske, F.; Hamzi, B. Kernel-Based Approximation of the Koopman Generator and Schrödinger Operator. *Entropy* **2020**, *22*, 722. [CrossRef] [PubMed]
32. Burov, D.; Giannakis, D.; Manohar, K.; Stuart, A. Kernel Analog Forecasting: Multiscale Test Problems. *Multiscale Model. Simul.* **2021**, *19*, 1011–1040. [CrossRef]
33. Baddoo, P.J.; Herrmann, B.; McKeon, B.J.; Brunton, S.L. Kernel Learning for Robust Dynamic Mode Decomposition: Linear and Nonlinear Disambiguation Optimization. *Proc. R. Soc.* **2022**, *478*, 20210830. [CrossRef]
34. Brunton, S.L.; Brunton, B.W.; Proctor, J.L.; Kutz, J.N. Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control. *PLoS ONE* **2016**, *11*, e0150171. [CrossRef]
35. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 3932–3937. [CrossRef] [PubMed]
36. Eckmann, J.P.; Ruelle, D. Ergodic Theory of Chaos and Strange Attractors. *Rev. Mod. Phys.* **1985**, *57*, 617–656. [CrossRef]
37. Zhu, R.; Cao, Y.; Kang, Y.; Wang, X. The Deep Input-Koopman Operator for Nonlinear Systems. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, 13–16 December 2018, Proceedings, Part VII 25*; Springer: Cham, Switzerland, 2018.
38. Lusch, B.; Kutz, J.N.; Brunton, S.L. Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics. *Nat. Commun.* **2018**, *9*, 4950. [CrossRef] [PubMed]
39. Qian, S.; Chou, C.A. A Koopman-operator-theoretical Approach for Anomaly Recognition and Detection of Multi-Variate EEG System. *Biomed. Signal Process. Control* **2021**, *69*, 102911. [CrossRef]
40. Ishikawa, I.; Hashimoto, Y.; Ikeda, M.; Kawahara, Y. Koopman operators with intrinsic observables in rigged reproducing kernel Hilbert spaces. *arXiv* **2024**, arXiv:2403.02524.
41. Bevanda, P.; Sosnowski, S.; Hirche, S. Koopman Operator Dynamical Models: Learning, Analysis and Control. *Annu. Rev. Control* **2021**, *52*, 197–212. [CrossRef]
42. Xiong, W.; Huang, X.; Zhang, Z.; Deng, R.; Sun, P.; Tian, Y. Koopman Neural Operator as a Mesh-Free Solver of Non-Linear Partial Differential Equations. *arXiv* **2023**, arXiv:2301.10022. https://doi.org/10.48550/arXiv.2301.10022.
43. Xiong, W.; Ma, M.; Huang, X.; Zhang, Z.; Sun, P.; Tian, Y. KoopmanLab: Machine Learning for Solving Complex Physics Equations. *arXiv* **2023**, arXiv:2301.01104. https://doi.org/10.48550/arXiv.2301.01104.
44. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]

45. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv* **2020**, arXiv:2010.08895.
46. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural Operator: Graph Kernel Network for Partial Differential Equations. *arXiv* **2020**, arXiv:2003.03485. https://doi.org/10.48550/arXiv.2003.03485.
47. Lu, L.; Jin, P.; Karniadakis, G.E. DeepONet: Learning Nonlinear Operators for Identifying Differential Equations Based on the Universal Approximation Theorem of Operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [CrossRef]
48. Shafienya, H.; Regan, A.C. 4D flight trajectory prediction using a hybrid Deep Learning prediction method based on ADS-B technology: A case study of Hartsfield–Jackson Atlanta International Airport (ATL). *Transp. Res. Part C Emerg. Technol.* **2022**, *144*, 103878. [CrossRef]
49. Jia, P.; Chen, H.; Zhang, L.; Han, D. Attention-LSTM based prediction model for aircraft 4-D trajectory. *Sci. Rep.* **2022**, *12*, 15533. [CrossRef] [PubMed]
50. Choi, H.C.; Deng, C.; Hwang, I. Hybrid machine learning and estimation-based flight trajectory prediction in terminal airspace. *IEEE Access* **2021**, *9*, 151186–151197. [CrossRef]
51. Packard, N.H.; Crutchfield, J.P.; Farmer, J.D.; Shaw, R.S. Geometry from a time series. *Phys. Rev. Lett.* **1980**, *45*, 712. [CrossRef]
52. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *arXiv* **2021**, arXiv:2012.07436. https://doi.org/10.48550/arXiv.2012.07436.
53. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. *arXiv* **2018**, arXiv:1703.07015. https://doi.org/10.48550/arXiv.1703.07015.
54. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
55. Lu, J.; Chai, H.; Jia, R. A General Framework for Flight Maneuvers Automatic Recognition. *Mathematics* **2022**, *10*, 1196. [CrossRef]
56. Liu, Y.; Li, C.; Wang, J.; Long, M. Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors. *arXiv* **2023**, arXiv:2305.18803. https://doi.org/10.48550/arXiv.2305.18803.
57. Wang, R.; Dong, Y.; Arik, S.Ö.; Yu, R. Koopman neural forecaster for time series with temporal distribution shifts. *arXiv* **2022**, arXiv:2210.03675.
58. Zhou, T.; Ma, Z.; Wen, Q.; Sun, L.; Yao, T.; Yin, W.; Jin, R. Film: Frequency improved legendre memory model for long-term time series forecasting. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 12677–12690.
59. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In Proceedings of the Eleventh International Conference on Learning Representations, Virtual, 25–29 April 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.