



PAPER • OPEN ACCESS

Time-delayed transformers for data-driven modeling of low-dimensional dynamics

To cite this article: Albert Alcalde *et al* 2026 *Mach. Learn.: Sci. Technol.* **7** 045038

View the [article online](#) for updates and enhancements.

You may also like

- [Adaptive hierarchical feature extraction with transformer-augmented CNNs for high-dimensional fluid mechanics data analysis](#)
Wenkai Li and Yulin Zhou
- [Forecasting high-dimensional spatio-temporal systems from sparse measurements](#)
Jialin Song, Zezheng Song, Pu Ren et al.
- [Machine-learned closure of URANS for stably stratified turbulence: connecting physical timescales & data hyperparameters of deep time-series models](#)
Muralikrishnan Gopalakrishnan Meena, Demetri Liousas, Andrew D Simin et al.



PAPER

OPEN ACCESS

RECEIVED
2 April 2026REVISED
3 July 2026ACCEPTED FOR PUBLICATION
21 July 2026PUBLISHED
4 August 2026

Original content from
this work may be used
under the terms of the
Creative Commons
Attribution 4.0 licence.

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Time-delayed transformers for data-driven modeling of low-dimensional dynamics

Albert Alcalde^{1,*} , Markus Widhalm² and Emre Yilmaz²¹ Chair for Dynamics, Control, Machine Learning, and Numerics (Alexander von Humboldt Professorship), Friedrich–Alexander–Universität Erlangen–Nürnberg, 91058 Erlangen, Germany² Institute of Aerodynamics and Flow Technology, DLR, 38108 Braunschweig, Germany

* Author to whom any correspondence should be addressed.

E-mail: albert.alcalde@fau.de**Keywords:** transformers, self-attention mechanism, scientific machine learning

Abstract

We propose the time-delayed transformer (TD-TF), a simplified TF architecture for data-driven modeling of unsteady spatio-temporal dynamics. TD-TF bridges linear operator-based methods and deep sequence models by showing that a single-layer, single-head TF can be interpreted as a nonlinear generalization of TD dynamic mode decomposition. The architecture is deliberately minimal, consisting of one self-attention layer with a single query per prediction and one feedforward layer, resulting in linear computational complexity in sequence length and a small parameter count. Numerical experiments demonstrate that TD-TF matches the performance of linear baselines on near-linear systems, while significantly outperforming them in nonlinear and chaotic regimes, where it accurately captures long-term dynamics. Validation studies on synthetic signals, unsteady aerodynamics, the Lorenz'63 system, and a reaction-diffusion model show that TD-TF preserves the interpretability and efficiency of linear models while providing substantially enhanced expressive power for complex dynamics.

1. Introduction

The availability of increasingly large high-resolution datasets, both in space and time, together with recent advances in scientific machine learning (SciML), is transforming how we model the evolution of physical systems. In particular, the modeling of unsteady flows and other high-dimensional dynamical systems has benefited from such data-centric approaches, which complement or extend classical methods derived from first principles.

Contrary to the traditional *physics-driven* paradigm, where one derives a system of ordinary differential equations (ODEs) from governing physical laws, the *data-driven* approach [11, 33] aims to identify an evolution operator that reproduces the observed dynamics directly from data. Even when the measurements are partial or low-dimensional, time-delay embedding techniques [40] allow for reconstructing the underlying attractor and recovering temporal correlations. Koopman operator theory [23, 32] provides a linear framework for representing nonlinear systems, and serves as the theoretical basis of many modern data-driven models. Within this family, dynamic mode decomposition (DMD) [35] offers a simple yet effective approach to extract the best-fit linear dynamical system describing the temporal evolution of observables. By enriching the state with delayed snapshots, time-delayed DMD (TD-DMD) captures higher-order temporal structures [25, 41]. However, TD-DMD remains linear, which fundamentally limits its ability to capture strongly nonlinear or chaotic behavior commonly encountered in physical systems.

At the other end of the modeling spectrum, transformer (TF) architectures [43] have emerged as powerful nonlinear sequence models built from feedforward neural networks [12, 17] and self-attention mechanisms [43]. Originally developed for natural language processing, TFs have since been adapted to time-series forecasting [48, 52] and operator learning [9, 10, 18] in SciML. The lack of transparency and

the quadratic complexity of full attention can be problematic in engineering contexts, where understanding the model and scaling to long sequences are important. By using self-attention to weight past information, TFs can learn complex dependencies across multiple time scales. These models achieve remarkable expressiveness on challenging tasks such as turbulent flow prediction and the construction of surrogate models for partial differential equations (PDEs) [16, 37, 49]. However, general-purpose TFs typically involve deep, multi-layer architectures that sacrifice interpretability and efficiency.

The tension between interpretability and expressivity is particularly evident in unsteady aerodynamics [4]. Short-term atmospheric disturbances, known as gust encounters, can cause rapid changes in lift and drag, potentially leading to structural overstressing or loss of control in flight [14]. During aircraft design and certification, large ensembles of such unsteady scenarios must be analyzed to ensure compliance with safety standards [15]. High-fidelity CFD simulations are often prohibitively expensive due to the nonlinear phenomena involved (e.g. shocks and flow separation), while simplified linearized models lack accuracy in strongly unsteady regimes [1, 44]. These challenges highlight the need for efficient yet interpretable data-driven models capable of learning nonlinear temporal responses from limited training data. Motivated by this challenge, we hypothesize that introducing structured nonlinearity into TD models can bridge the gap between linear methods like TD-DMD and black-box sequence models.

Our approach proposes a new structured TF architecture, the TD-TF. TD-TF is deliberately designed to mirror the form of TD-DMD while extending it with nonlinear expressive capacity. In our formulation, the model's prediction at each time step can be interpreted as the action of a learned evolution operator on a finite window of past states. In concrete terms, TD-TF produces the next state as a weighted sum of nonlinear transformations of the past n states: each past snapshot is first passed through a feedforward network which acts as a learned nonlinear *feature map*, and then combined via attention weights that depend on their relevance to the latest state. This preserves the central assumption of TD-DMD that future states depend only on a finite recent history, but relaxes the linearity constraint, allowing those past states to interact in a nonlinear manner.

1.1. Our contributions

This work introduces a new perspective on unifying linear delay-based models and TFs. Our main contributions are:

- **Nonlinear reinterpretation of TD-DMD.** We demonstrate that a single-layer, single-head self-attention model can be interpreted as a nonlinear generalization of TD-DMD. This provides a direct theoretical bridge between our TF architecture and classical data-driven system identification techniques.
- **Novel TD-TF architecture.** We propose a new architecture that enforces a TD-DMD-inspired structure, using one feedforward transformation per state and one attention query per prediction, yielding linear complexity in sequence length. Positional information is encoded by concatenating time indices rather than learned embeddings, further simplifying the model.
- **Numerical validation.** We validate TD-TF on a range of systems from simple linear to highly nonlinear and chaotic. These include (i) sinusoidal signals where an exact linear solution exists, (ii) an unsteady aerodynamics problem (gust response of an airfoil) with high-dimensional flow dynamics close to a linear regime, (iii) the Lorenz'63 chaotic system, and (iv) a reaction–diffusion partial differential equation projected onto a low-dimensional basis. Our experiments show that TD-TF achieves comparable accuracy to TD-DMD in or close to linear regimes, and improves upon TD-DMD in strongly nonlinear and chaotic regimes.

1.2. Related work

Learning temporal dynamics from data has been studied extensively, from classical linear identification methods [42] to modern neural operator models [30]. We briefly review the main approaches most relevant to this work.

Data-driven dynamical modeling. A large body of work focuses on identifying system dynamics from data. Classical linear techniques such as DMD [35] and its extensions, such as extended and kernel DMD [46, 47], approximate the Koopman operator [23, 32] to obtain linear representations of nonlinear dynamics. Time-delay embeddings [40] enable reconstruction of partially observed systems and lead to methods such as TD-DMD [25, 41] and HAVOK [6]. Equation discovery approaches such as sparse identification of nonlinear dynamics [7, 34] recover governing equations through sparse regression, whose interpretability and performance are tied to selecting a proper set of candidate terms for the library.

TFs for sequence learning. TFs have been applied to time-series forecasting tasks in many domains. Recent studies draw parallels between self-attention and autoregressive (AR) models [29, 50], where the TF's attention mechanism with a causal mask is seen as a data-driven generalization of a linear vector AR process [31]. This connection has been used to interpret TFs as nonlinear AR models and to explore their ability to capture long-range dependencies. More recently, architectures such as the Delayformer have been proposed to model spatiotemporal dynamics by explicitly integrating temporal transformations for multivariate prediction, reflecting growing interest in attention-based models tailored to dynamics [45]. Relatedly, recent work has adopted an operator-learning viewpoint and shown that TF models can learn transfer operators in context [3]. In contrast to these works, our formulation of the TD-TF explicitly enforces an AR structure by attending only from the last input state to previous states, rather than attending over all pairwise interactions.

TFs for physical systems. Recently, neural operators such as the Fourier Neural Operator [24, 26] and DeepONet [30] have been proposed for learning solution operators of physical systems governed by PDEs. Recent developments further connect attention, state-space sequence models, and neural-operator ideas for evolving physical systems, including TF-based forecasting from sparse measurements [38], attention-based spatio-temporal neural operators [22], and state-space neural operators for dynamical systems [21]. TFs have been adapted to this setting due to their scalability and capacity to model long-range dependencies, and have demonstrated strong performance relative to convolutional or recurrent architectures on complex PDEs [9, 10, 19]. Our work differs by having the objective of retaining interpretability by using a single self-attention layer and a constrained architecture, thus avoiding the black-box nature of deep TFs.

1.3. Organization of the paper

The remainder of the paper is organized as follows. In section 2, we formalize the problem setting. In section 3, we introduce and motivate the two delay-based models considered in this work, emphasizing their connections and distinctive features. In section 4, we present numerical results on four case studies of increasing complexity, comparing TD-TF with TD-DMD. Finally, we summarize our conclusions in section 5.

2. Problem setting

We consider an autonomous dynamical system described by a system of ODEs:

$$\frac{d\mathbf{w}(t)}{dt} = \mathbf{f}(\mathbf{w}(t)), \quad \mathbf{w}(0) = \mathbf{w}_0, \quad (1)$$

where the vector-valued function $\mathbf{f}: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is unknown. However, we can observe $\mathbf{w} \in \mathbb{R}^d$ using a set of N available trajectories. Along a sequence of equally spaced time instances, $t_k = k\Delta t$, where $k = 0, \dots, K$, we denote the state at time t_k along the i -th trajectory with initial condition $\mathbf{w}_0^i$ as

$$\mathbf{w}_k^i = \mathbf{w}(t_k; \mathbf{w}_0^i), \quad \text{where } k = 0, \dots, K \quad \text{and} \quad i = 1, \dots, N. \quad (2)$$

Our objective is to create a data-driven model that predicts the evolution of $\mathbf{w}$ from any initial state $\mathbf{w}_0^i$. In many real-world scenarios, $\mathbf{w}$ is a low-dimensional observable of a large (and possibly infinite) dimensional system, so we are particularly interested in dynamics where d is relatively small.

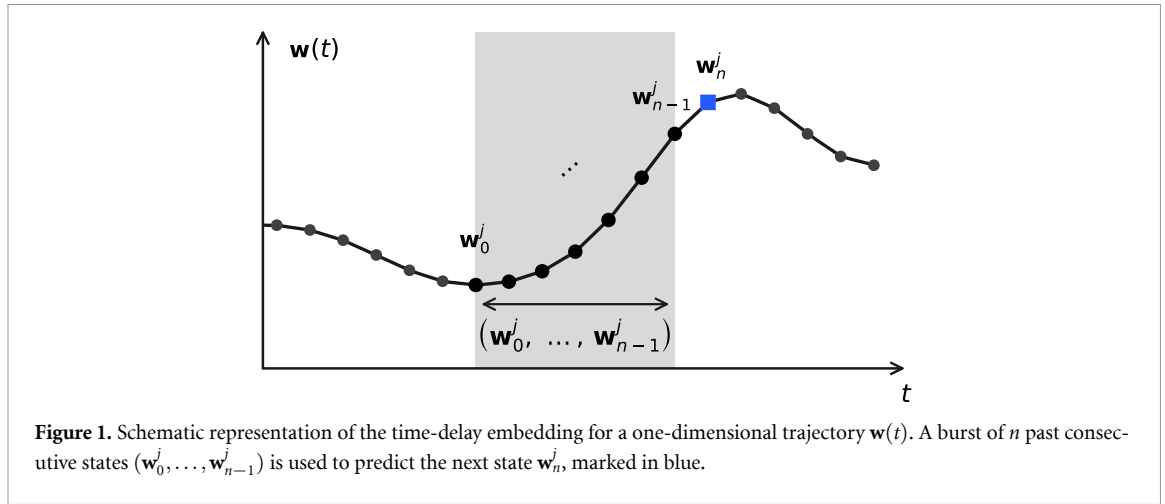
Furthermore, we relax the strong requirement of having access to the observable for all time instances. Following the framework of [11], we assume access to only bursts of consecutive time instances randomly spaced along the full trajectories. From this dataset, the dynamics of the observable $\mathbf{w}$ can be reconstructed in the following way.

First, we denote the J bursts of $n+1$ consecutive states in time (randomly sampled from the N total trajectories) by

$$(\mathbf{w}_0^j, \dots, \mathbf{w}_n^j), \quad j = 1, \dots, J. \quad (3)$$

Then, a delay-based model, $\mathbf{F}: \mathbb{R}^{nd} \rightarrow \mathbb{R}^d$, is built by incorporating information from the previous n states to predict the next state

$$\mathbf{w}_n^j \approx \mathbf{F}\left(\left[\mathbf{w}_0^j; \dots; \mathbf{w}_{n-1}^j\right]\right), \quad j = 1, \dots, J. \quad (4)$$



An example of this time-delay embedding of a one-dimensional observable is illustrated in figure 1. Many physical phenomena fit this problem formulation, such as two-dimensional flow past a cylinder, for which we can measure the lift coefficient, $C_l \in \mathbb{R}$, along certain time windows. In this paper, we consider the scenario of measuring several trajectories of an observable, with the goal of reconstructing trajectories for arbitrary times and initial conditions.

3. Delay-based data-driven modeling

In this section, we introduce the two data-driven methods considered in this work, namely, (i) TD-DMD and (ii) the TD-TF. In particular, we highlight how these two methods are related to each other.

3.1. TD-DMD

We start by briefly reviewing DMD [35]. The core idea behind this method is to find the best linear approximation such that

$$w_{k+1}^j \approx A w_k^j \quad (5)$$

holds for all times $k = 0, \dots, n-1$ and all $j = 1, \dots, J$, where the linear mapping is represented by the matrix $A \in \mathbb{R}^{d \times d}$. For a compact representation, we denote X and Y as follows:

$$X = \begin{bmatrix} w_0^1 & \dots & w_{n-1}^1 & | & \dots & | & w_0^j & \dots & w_{n-1}^j \end{bmatrix} \in \mathbb{R}^{d \times nj}, \quad (6)$$

$$Y = \begin{bmatrix} w_1^1 & \dots & w_n^1 & | & \dots & | & w_1^j & \dots & w_n^j \end{bmatrix} \in \mathbb{R}^{d \times nj}. \quad (7)$$

In this case, we assume that the approximation $Y \approx AX$ holds, for which the minimum-residual solution in the Frobenius norm is

$$\arg \min_A \|Y - AX\|_F^2 = \arg \min_A \sum_{j=1}^J \sum_{k=0}^{n-1} \|w_{k+1}^j - A w_k^j\|^2. \quad (8)$$

An approximate solution for the minimization problem corresponding to (8) can be attained by employing the following expression:

$$A = YX^\dagger = YV\Sigma^\dagger U^*, \quad (9)$$

where $\dagger$ denotes the Moore–Penrose inverse or *pseudoinverse*, and $X = U\Sigma V^*$ is the singular value decomposition (SVD) of X . Given an initial state $w_0 \in \mathbb{R}^d$ and a trajectory index j , the dynamics can be reconstructed as

$$w_k^j = A w_{k-1}^j = A^k w_0^j \quad (10)$$

for all $k \geq 0$.

A well-known shortcoming of DMD is that it might fail to reconstruct complex dynamics when the dimension of the observables is relatively low. The scalar case, $d = 1$, is particularly restrictive, since the DMD matrix, $\mathbf{A} = a \in \mathbb{R}$, boils down to a scalar. The model then becomes

$$\mathbf{w}_{k+1} = a\mathbf{w}_k, \quad k \geq 0, \quad (11)$$

with solution $\mathbf{w}_k = a^k \mathbf{w}_0$. Thus, only single-exponential dynamics can be represented, so multi-rate or more complex temporal behavior cannot be captured.

TD-DMD [2] seeks to overcome this problem by enlarging the states to a prescribed number of time-delays, $n \geq 0$, an approach based on Takens' embedding theorem [40]. The idea is that the following sum of linear approximations

$$\mathbf{w}_n^j \approx \mathbf{A}_{n-1} \mathbf{w}_{n-1}^j + \dots + \mathbf{A}_0 \mathbf{w}_0^j \quad (12)$$

is assumed to hold approximately for all trajectories $j = 1, \dots, J$. More compactly, we define augmented states for all pairs of indices $i \leq k$ as

$$\mathbf{w}_{i:k}^j = \begin{bmatrix} \mathbf{w}_i^j \\ \vdots \\ \mathbf{w}_k^j \end{bmatrix} \in \mathbb{R}^{d(k-i+1)} \quad (13)$$

and build the matrices of the augmented states as

$$\mathbf{X} = [\mathbf{w}_{0:n-1}^1 \quad \dots \quad \mathbf{w}_{0:n-1}^J] \in \mathbb{R}^{dn \times J} \quad (14)$$

$$\mathbf{Y} = [\mathbf{w}_{1:n}^1 \quad \dots \quad \mathbf{w}_{1:n}^J] \in \mathbb{R}^{dn \times J}. \quad (15)$$

If $\mathbf{Y} \approx \mathbf{A}_{\text{TD}} \mathbf{X}$ holds, $\mathbf{A}_{\text{TD}}$ has the following structure for all $j = 1, \dots, J$:

$$\underbrace{\begin{bmatrix} \mathbf{w}_1^j \\ \mathbf{w}_2^j \\ \vdots \\ \mathbf{w}_{n-1}^j \\ \mathbf{w}_n^j \end{bmatrix}}_{\mathbf{w}_{1:n}^j} \approx \underbrace{\begin{bmatrix} \mathbf{0}_d & \mathbf{I}_d & \mathbf{0}_d & \dots & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{0}_d & \mathbf{I}_d & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0}_d \\ \mathbf{0}_d & \dots & \mathbf{0}_d & \mathbf{0}_d & \mathbf{I}_d \\ \mathbf{A}_0 & \dots & \dots & \mathbf{A}_{n-2} & \mathbf{A}_{n-1} \end{bmatrix}}_{\mathbf{A}_{\text{TD}}} \underbrace{\begin{bmatrix} \mathbf{w}_0^j \\ \mathbf{w}_1^j \\ \vdots \\ \mathbf{w}_{n-2}^j \\ \mathbf{w}_{n-1}^j \end{bmatrix}}_{\mathbf{w}_{0:n-1}^j}. \quad (16)$$

Here, $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ and $\mathbf{0}_d \in \mathbb{R}^{d \times d}$ denote the identity matrix and the matrix full of zeros, respectively. As in DMD, the best-fit linear operator is obtained in the minimal Frobenius norm sense as $\mathbf{A}_{\text{TD}} = \mathbf{Y}\mathbf{X}^\dagger$. The matrix that contains the last d rows of $\mathbf{A}_{\text{TD}}$ can be denoted by

$$\hat{\mathbf{A}}_{\text{TD}} = [\mathbf{A}_0 \quad \dots \quad \mathbf{A}_{n-1}] \in \mathbb{R}^{d \times nd}. \quad (17)$$

Then, the future evolution of an initial augmented state, $\mathbf{w}^j \in \mathbb{R}^{nd}$, can be generated by recursively applying

$$\mathbf{w}_k^j = \hat{\mathbf{A}}_{\text{TD}} \mathbf{w}_{k-n:k-1}^j \quad (18)$$

for all $k \geq n$.

The advantages of TD-DMD are that it is computationally cheap, only requiring SVD and $O(n)$ operations at evaluation time, as well as being rooted in Koopman operator theory [41]. The main limitation stems from the fact that it is a linear approximation. Therefore, it might struggle to capture complex nonlinear dynamics. Furthermore, the number of parameters grows with the number of delays n , which is also a limiting factor. We address these limitations of the TD-DMD approach by introducing TD TFs, inspired by the TF architecture and its similarities to TD-DMD.

3.2. TD-TFs

This section introduces the main components of the TF architecture that we use to build TD-TFs, namely, positional encoding, feedforward layers, and self-attention layers.

Positional encoding. It is possible that identical state values occur at different time instances, which would prevent the model from distinguishing between them. *Positional encoding* tackles this issue by including explicit temporal information in each state. In our particular implementation, we simply append to each state its normalized time index. This yields

$$\mathbf{y}_k^j = \mathbf{PE}(\mathbf{w}_k^j) := \begin{bmatrix} \mathbf{w}_k^j \\ k/n \end{bmatrix} \in \mathbb{R}^{d_{\text{in}}}, \quad \text{for all } k = 0, \dots, n-1, \quad (19)$$

where $d_{\text{in}} = d + 1$. This choice is deliberately minimal: self-attention without positional information is permutation equivariant, and appending the normalized time index breaks this symmetry while avoiding the additional architectural complexity of sinusoidal or learned positional embeddings [43, 51].

Feedforward layer. The first key element of a TF is the feedforward layer. This is defined as a map $\mathbf{FF} : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}^{d_{\text{in}}}$ shared across time steps, and is parametrized as a shallow fully-connected neural network. For a given state, $\mathbf{y} \in \mathbb{R}^{d_{\text{in}}}$, this mapping can be expressed as

$$\mathbf{FF}(\mathbf{y}) = \mathbf{W}\sigma(\mathbf{U}\mathbf{y} + \mathbf{b}), \quad (20)$$

where $\sigma : \mathbb{R}^h \rightarrow \mathbb{R}^h$ is a nonlinear activation function, $\mathbf{W} \in \mathbb{R}^{d_{\text{in}} \times h}$, $\mathbf{U} \in \mathbb{R}^{h \times d_{\text{in}}}$ are the matrices of the two consecutive linear layers, and $\mathbf{b} \in \mathbb{R}^h$ is the bias of the first linear layer for the hidden dimension $h \in \mathbb{N}$.

Self-attention layer. The second key element of a TF is a self-attention layer, defined as the map $\mathbf{SA} : \mathbb{R}^{d_{\text{in}} n} \rightarrow \mathbb{R}^d$ from a collection of n vectors in $\mathbb{R}^{d_{\text{in}}}$ to a vector in $\mathbb{R}^d$. Specifically, for a given input, $\mathbf{y}_{0:n-1} = [\mathbf{y}_0; \dots; \mathbf{y}_{n-1}] \in \mathbb{R}^{d_{\text{in}} n}$, this mapping can be represented as

$$\mathbf{SA}(\mathbf{y}_{0:n-1}) = \sum_{k=0}^{n-1} \alpha_{n-1,k}(\mathbf{y}_{0:n-1}) \mathbf{V}\mathbf{y}_k \quad (21)$$

where

$$\alpha_{n-1,k}(\mathbf{y}_{0:n-1}) = \frac{e^{\langle \mathbf{y}_{n-1}, \mathbf{B}\mathbf{y}_k \rangle}}{\sum_{r=0}^{n-1} e^{\langle \mathbf{y}_{n-1}, \mathbf{B}\mathbf{y}_r \rangle}} \in (0, 1). \quad (22)$$

Here, $\mathbf{B} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ and $\mathbf{V} \in \mathbb{R}^{d \times d_{\text{in}}}$ are parameter matrices, known as the *query-key* matrix and the *value* matrix, respectively. It is worth noting that the function $\mathbf{SA}$ can be applied directly to the augmented states, $\mathbf{y}_{0:n-1}^j \in \mathbb{R}^{d_{\text{in}} n}$, or to their transformation after applying the feedforward layer component-wise. We also note that the attention coefficients, $\alpha_{n-1,k}(\mathbf{y}_{0:n-1})$, in (22) depend on the pairwise inner-products between the last state, $\mathbf{y}_{n-1}$, and all states in $\mathbf{y}_{0:n-1}$.

Building the TD TF. We define the TF model by composing positional encoding, a feedforward layer, and a self-attention layer. Specifically, the TF model is defined as the mapping $\mathbf{TF} : \mathbb{R}^{d_{\text{in}} n} \rightarrow \mathbb{R}^d$. For a given input already with positional encoding $[\mathbf{y}_0; \dots; \mathbf{y}_{n-1}] \in \mathbb{R}^{d_{\text{in}} n}$, this mapping is represented as

$$\mathbf{TF}([\mathbf{y}_0; \dots; \mathbf{y}_{n-1}]) = \mathbf{SA}([\mathbf{FF}(\mathbf{y}_0); \dots; \mathbf{FF}(\mathbf{y}_{n-1})]). \quad (23)$$

We provide a pseudocode description of the TD-TF approach in algorithm 1. The main benefit of the TD-TF is that it is nonlinear and independent of the number of delays, n , in the parameter count, since the same feature maps are used for all delays. However, this comes at the cost of $\mathcal{O}(n)$ operations in the self-attention layer. Therefore, the nonlinearities arising from both the feature maps and the attention weights overcome the limitations of an inherently linear model such as TD-DMD at the cost of expensive training of the parameters in its layers.

Algorithm 1. Time-delayed transformer.

Require: States $\{\mathbf{w}_k\}_{k=0}^{n-1} \subset \mathbb{R}^d$; $\text{flag pos_enc} \in \{\text{True}, \text{False}\}$
Ensure: Output $\mathbf{o} \in \mathbb{R}^d$

```

1: function TD-TF( $\{\mathbf{w}_k\}_{k=0}^{n-1}, \text{pos\_enc}$ )
2:   if pos_enc then
3:      $d_{\text{in}} \leftarrow d + 1$ 
4:   else
5:      $d_{\text{in}} \leftarrow d$ 
6:   end if
Parameters:  $\mathbf{U} \in \mathbb{R}^{h \times d_{\text{in}}}$ ,  $\mathbf{W} \in \mathbb{R}^{d_{\text{in}} \times h}$ ,  $\mathbf{b} \in \mathbb{R}^h$ ,  $\mathbf{B} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ ,  $\mathbf{V} \in \mathbb{R}^{d \times d_{\text{in}}}$ ; nonlinearity  $\sigma : \mathbb{R}^h \rightarrow \mathbb{R}^h$ 
7:   for  $k = 0, \dots, n-1$  do
8:     if pos_enc then
9:        $\mathbf{y}_k \leftarrow \begin{bmatrix} \mathbf{w}_k \\ k/n \end{bmatrix} \in \mathbb{R}^{d+1}$ 
10:    else
11:       $\mathbf{y}_k \leftarrow \mathbf{w}_k \in \mathbb{R}^d$ 
12:    end if
13:     $\mathbf{z}_k \leftarrow \mathbf{W}\sigma(\mathbf{U}\mathbf{y}_k + \mathbf{b})$ 
14:  end for
15:  for  $k = 0, \dots, n-1$  do
16:     $s_k \leftarrow \langle \mathbf{z}_{n-1}, \mathbf{B}\mathbf{z}_k \rangle$ 
17:  end for
18:   $\alpha_{n-1,k} \leftarrow \exp(s_k) / \sum_{r=0}^{n-1} \exp(s_r)$  for  $k = 0, \dots, n-1$ 
19:  return  $\mathbf{o} \leftarrow \sum_{k=0}^{n-1} \alpha_{n-1,k} \mathbf{V}\mathbf{z}_k$ 
20: end function

```

Training strategy and evaluation. We now describe how the TD-TF is applied to the problem setting described in section 2. Inspired by residual formulations of neural networks [17], we include an identity mapping so that the TD-TF learns state increments. For this, we approximate

$$\mathbf{w}_n^j - \mathbf{w}_{n-1}^j \approx \text{TF} \left(\begin{bmatrix} \mathbf{w}_0^j; \dots; \mathbf{w}_{n-1}^j \end{bmatrix} \right) \quad (24)$$

for all $j = 1, \dots, J$. Unlike the TD-DMD approach, for which a closed form solution exists, we need to solve an optimization problem to obtain the model parameters in this case. Next, we denote all parameters in TD-TF as $\theta = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{B}, \mathbf{V}\}$, which can be learned by solving the following minimization problem:

$$\arg \min_{\theta} \frac{1}{J} \sum_{j=1}^J \left\| \mathbf{w}_n^j - \mathbf{w}_{n-1}^j - \text{TF}_{\theta} \left(\begin{bmatrix} \mathbf{w}_0^j; \dots; \mathbf{w}_{n-1}^j \end{bmatrix} \right) \right\|^2. \quad (25)$$

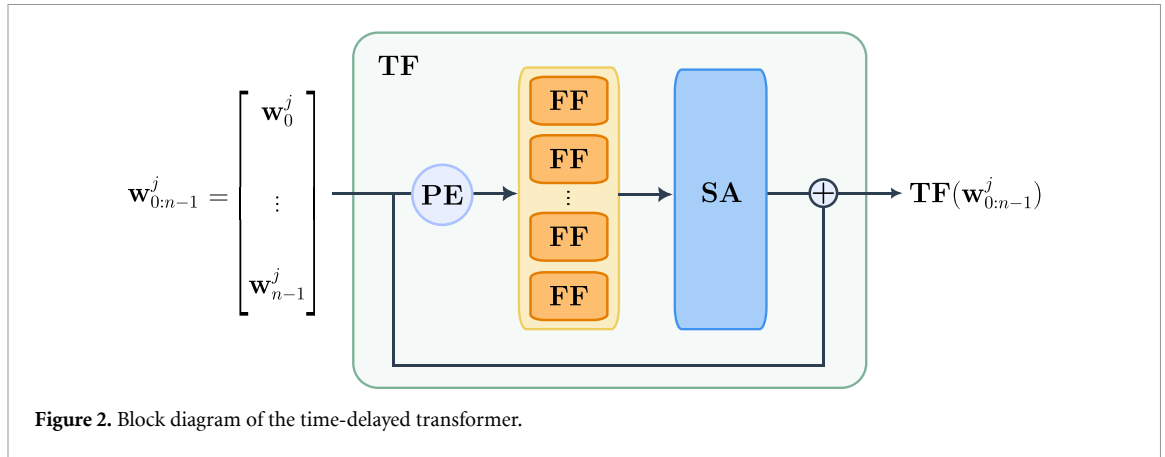
As is common with neural network training, we use a stochastic gradient-based algorithm, AdamW [28]. After training the TD-TF model, it can iteratively be used to reconstruct the dynamics of any initial augmented state $\mathbf{w}_{0:n-1}^j \in \mathbb{R}^{nd}$ by recursively applying

$$\mathbf{w}_k^j = \mathbf{w}_{k-1}^j + \text{TF} \left(\begin{bmatrix} \mathbf{w}_{k-n}^j; \dots; \mathbf{w}_{k-1}^j \end{bmatrix} \right) \quad \text{for all } k \geq n. \quad (26)$$

3.3. Connections to TF modeling and TD-DMD

TD-TF is a simplified variant of the standard TF architecture, designed to better align with the structure and interpretability of linear delay-based dynamical models. While conventional TFs were developed for high-dimensional sequence tasks such as language and vision, the TD-TF approach isolates the minimal architectural elements required for temporal prediction [48, 52] and reformulates them in a delay-based manner [35].

As illustrated in figure 2, the TD-TF model consists of a single feedforward layer applied to time-augmented states via positional encoding, followed by one self-attention layer. In contrast to the multi-layer, multi-head structure of standard TFs, we restrict the model to a single attention head and a single TF block. Motivated by the connection to TD-DMD, we modify the layer ordering by placing the feedforward layer before attention. We also omit normalization layers and replace intermediate residual



connections with a single residual link at the model level. Positional information is introduced not through sinusoidal [43] or learned embeddings [39], but by concatenating a normalized time coordinate to each state vector.

A key distinction of TD-TF lies in how attention is used for prediction. In standard TFs, self-attention maps $\mathbb{R}^{nd}$ to $\mathbb{R}^{nd}$, computing all pairwise interactions between states at $\mathcal{O}(n^2)$ cost. In contrast, TD-TF evaluates attention only at the final index to predict the next observable, $\mathbf{w}_n$, conditioned on the history $(\mathbf{w}_0, \dots, \mathbf{w}_{n-1})$. This reduces complexity to $\mathcal{O}(n)$, which can mitigate error accumulation during AR rollout. Conceptually, this design brings TD-TF closer to the structure of TD dynamical systems, where each new state is determined by a weighted combination of the past states. Moreover, since our problem setting involves fixed-length input sequences, the model need not generalize across variable-length contexts as in language modeling.

Finally, we establish the direct correspondence that emerges between the TD-TF and TD-DMD models. The TD-DMD update,

$$\mathbf{w}_n = \sum_{k=0}^{n-1} \mathbf{A}_k \mathbf{w}_k, \quad (27)$$

can be viewed as the linear analogue of the TD-TF mapping, $\mathbf{TF}(\mathbf{w}_{0:n-1})$, where each term $\mathbf{A}_k \mathbf{w}_k$ is replaced by a nonlinear, data-adaptive contribution,

$$\alpha_{n-1,k}(\mathbf{w}_{0:n-1}) \mathbf{VFF}(\mathbf{w}_k). \quad (28)$$

In this sense, TD-TF serves as a nonlinear, data-dependent generalization of TD-DMD, where the static matrices $\mathbf{A}_k$ are replaced by adaptive attention weights, and the feature map $\mathbf{FF}$ introduces nonlinear observables.

4. Numerical experiments

We evaluate the performance of the proposed TD-TF model against TD-DMD and an additional nonlinear neural baseline across several datasets of increasing dynamical complexity, namely, a simple periodic signal, an unsteady aerodynamic flow, a chaotic nonlinear system and a reaction-diffusion system. All datasets undergo identical preprocessing, where each time series is subsampled every $\tau \in \mathbb{N}$ steps and normalized to the hypercube $[-1, 1]^d$. Normalization is inverted at inference time.

The additional nonlinear baseline is a TD-MLP, which is introduced to distinguish the effect of the TD-TF attention mechanism from the effect of adding nonlinear sequence-modeling capacity. This model is designed to preserve the same delay window as TD-TF but without the attention mechanism. Given the delayed states $\{\mathbf{w}_k\}_{k=0}^{n-1}$, we first optionally apply the same positional encoding as in TD-TF and denote the corresponding input dimension by d_{in} . The TD-MLP is then defined as a shared nonlinear map $\tilde{\mathbf{FF}}: \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}^d$ applied independently to each delayed state, followed by summation over the delay window:

$$\mathbf{MLP}([\mathbf{y}_0; \dots; \mathbf{y}_{n-1}]) = \sum_{k=0}^{n-1} \tilde{\mathbf{FF}}(\mathbf{y}_k), \quad (29)$$

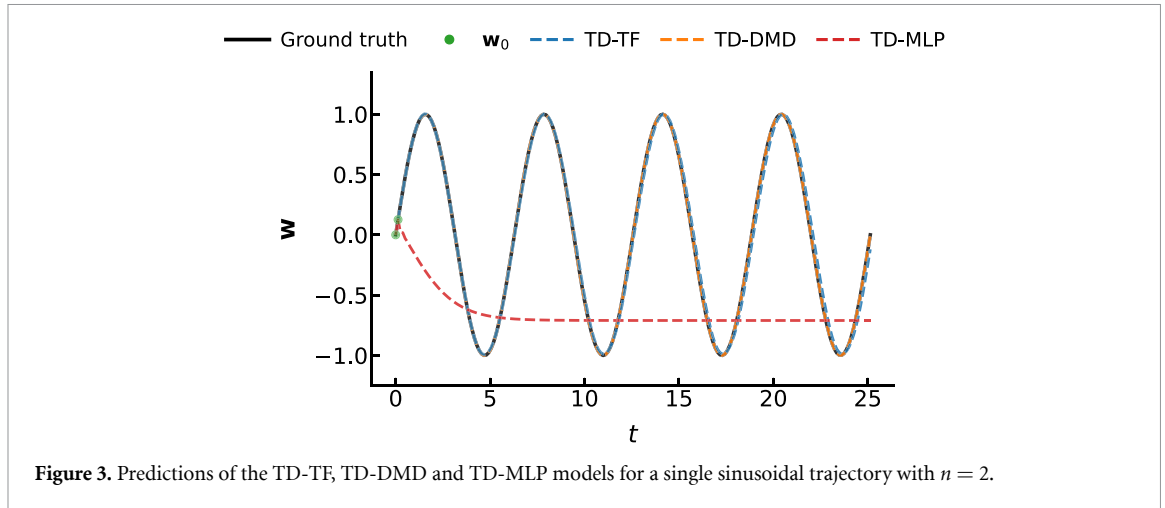


Table 1. RMSE for the sinusoidal trajectory with delay length $n = 2$. Neural network models use hidden dimension $h = 10$ and are evaluated over 5 random seeds.

	TD-TF	TD-DMD	TD-MLP
RMSE	$2.72 \cdot 10^{-1} \pm 3.92 \cdot 10^{-1}$	$9.31 \cdot 10^{-14}$	$9.36 \cdot 10^{-1} \pm 2.36 \cdot 10^{-3}$

The code used to generate the results can be found in the following repository: <https://github.com/albertalcalde/TDTF>.

4.1. Case 1: Sinusoidal trajectory

We consider a generic example of a single sinusoidal trajectory [5]. This analytically tractable case provides a baseline for assessing whether the models recover simple harmonic dynamics.

Setup. The observable states are defined by the discrete and scalar ($d = 1$) signal,

$$w_k = \sin(k\Delta t), \quad k \in \{0, \dots, K\}, \quad \Delta t = \frac{4\pi}{100}, \quad (30)$$

with $K = 200$, $J = 10$ bursts and $\tau = 1$ (i.e. no subsampling). This simple case serves as a benchmark where TD-DMD captures the dynamics for $n = 2$ exactly. Indeed, using trigonometric identities, one can show that perfect reconstruction is achieved with $A_1 = 2\cos(\Delta t)$ and $A_0 = -1$.

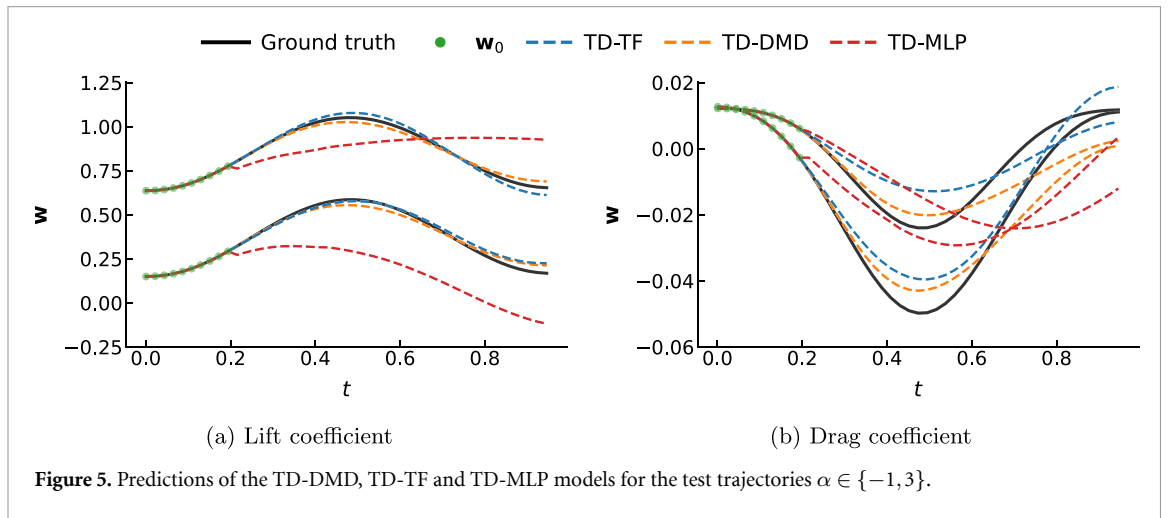
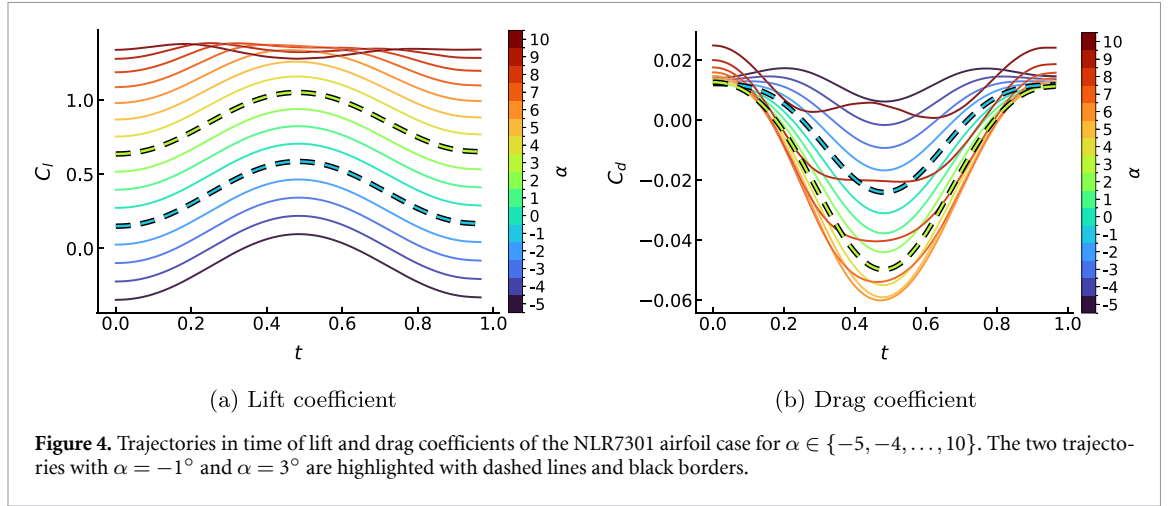
Results. For TD-TF, we select the hidden dimension as 10, the learning rate as 10^{-2} , and the batch size as 5, and we train for 1000 epochs. As shown in figure 3, TD-TF can reconstruct the dynamics with $n = 2$ delays, but with a noticeable phase shift as time increases further. This qualitative observation is reflected quantitatively by the root mean square error (RMSE): TD-DMD achieves a RMSE of $9.3 \cdot 10^{-14}$, corresponding to near machine-precision reconstruction, while TD-TF attains a RMSE of $4.8 \cdot 10^{-2}$.

We further compare TD-TF with the TD-MLP baseline using the same hidden dimension and number of delays, and evaluate their performance over 5 random seeds in table 1. As expected, TD-DMD reconstructs the sinusoidal trajectory to near machine precision, since the signal satisfies an exact second-order linear recurrence. In this linear setting, the nonlinear models do not improve upon TD-DMD, and TD-MLP has the largest error.

4.2. Case 2: Flow around the NLR7301 airfoil

We next consider unsteady flow around the NLR7301 airfoil [53] during a vertical gust encounter. This benchmark represents a high-dimensional, nonlinear, and time-dependent aerodynamic regime suitable for evaluating data-driven temporal models.

Setup. We generate the flow solutions and quantities of interest using the TAU code, the DLR in-house CFD solver [36], solving the unsteady Reynolds-averaged Navier–Stokes equations with a suitable turbulence closure. The unsteady forcing is introduced through a vertical ‘1 – cos’ gust profile following the discrete gust design criteria [15]. The discrete gust design criteria define scenarios by varying parameters including the angle of attack (AoA), free-stream Mach number (M_∞), Reynolds number (Re), gust amplitude (A_g), and gust length (l_g) within prescribed ranges.



For the present demonstration, we fix M_∞ at 0.35 and Re at $1.19 \cdot 10^6$, corresponding to a subsonic flow condition. The gust profile is characterized by a wavelength of 58.0 m, an amplitude of 8.0 m s^{-1} , and a horizontal offset of 0.2 m ahead of the airfoil nose. The angle of attack is varied in $\{-5, -4, \dots, 10\}$ degrees, yielding $N = 16$ trajectories of aerodynamic coefficients over the physical time domain of $[0, 0.966]$ seconds. Each trajectory is resolved with $K = 1349$ time steps and a temporal resolution of $\Delta t = 7.16 \cdot 10^{-4}$ seconds, and we set $J = 5000$ and $\tau = 30$.

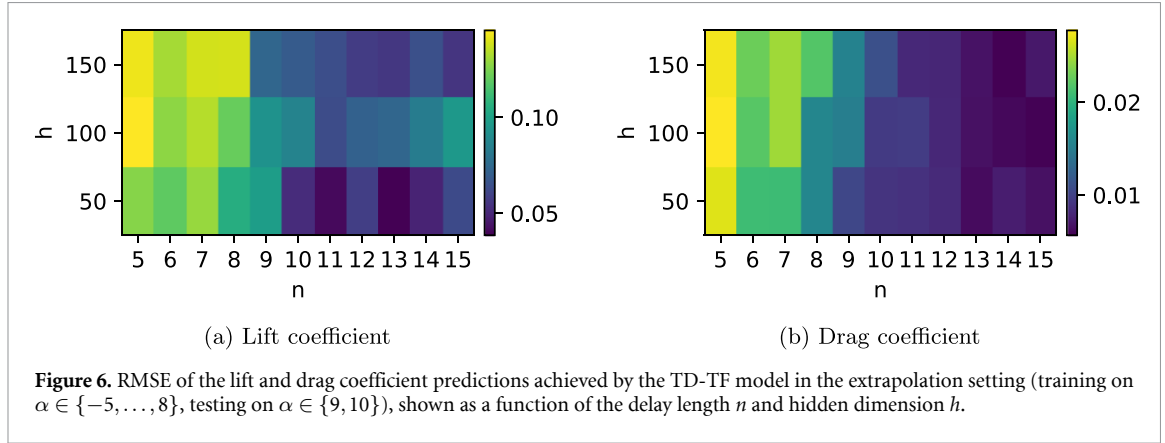
The quantities of interest are the lift and drag coefficients, denoted C_l and C_d , respectively. In all experiments below, we treat C_l and C_d as *separate scalar observables*: we train one model on the C_l time series and a second, independent model on the C_d time series (i.e. $d = 1$ in each case). These time series serve as ground truth and are shown in figure 4.

Results. As a first interpolation test, all trajectories except those corresponding to $\alpha = -1^\circ$ and $\alpha = 3^\circ$ are used as the training set; these two are selected because they fall well within the range of angles considered and are reserved for testing. We train the TD-TF and the TD-MLP models for 500 epochs with a hidden dimension of 100, learning rate of 10^{-2} , and batch size of 100. We present our results for $n = 10$ in figure 5.

For the lift coefficient, both the TD-DMD and TD-TF models achieve low average prediction error. In particular, TD-TF attains an average RMSE of $2.03 \cdot 10^{-2}$, with a maximum absolute error of $4.9 \cdot 10^{-2}$ over the prediction horizon. The drag coefficient presents a more challenging prediction task. TD-TF achieves an average RMSE of $5.57 \cdot 10^{-3}$ and a worst-case absolute error of $1.14 \cdot 10^{-2}$ over the test trajectories, with errors again increasing toward the end of the prediction horizon. Despite accurately capturing the overall dynamics, TD-TF systematically overestimates the minimum drag value: the predicted minimum is higher than the true minimum by $1.06 \cdot 10^{-2}$ on average. Moreover, the time at which the minimum drag occurs is shifted by approximately $3.2 \cdot 10^{-2}$ in nondimensional time, consistent with the visual discrepancy observed around $t \approx 0.5$ in figure 5.

Table 2. RMSE for the lift and drag trajectories with delay length $n = 10$. Both TD-TF and TD-MLP use hidden dimension $h = 100$ and are evaluated over 5 random seeds.

	TD-TF	TD-DMD	TD-MLP
RMSE (lift)	$9.40 \cdot 10^{-2} \pm 8.84 \cdot 10^{-2}$	$1.95 \cdot 10^{-2} \pm 2.71 \cdot 10^{-3}$	$1.80 \cdot 10^{-1} \pm 4.90 \cdot 10^{-2}$
RMSE (drag)	$7.85 \cdot 10^{-3} \pm 3.77 \cdot 10^{-3}$	$6.11 \cdot 10^{-3} \pm 1.17 \cdot 10^{-4}$	$1.44 \cdot 10^{-2} \pm 1.67 \cdot 10^{-3}$



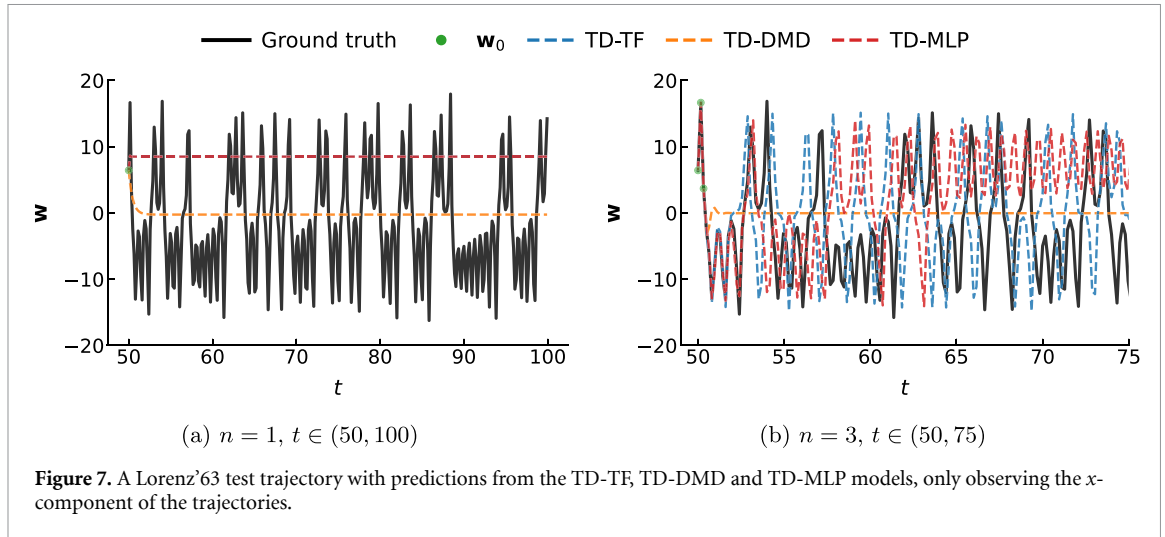
To assess robustness with respect to initialization, we report RMSE values averaged over 5 random seeds in table 2. The averaged results show that TD-MLP is less competitive in this example, yielding larger errors than both TD-DMD and TD-TF for both lift and drag. They also show that TD-TF is more sensitive to initialization than TD-DMD in this near-linear aerodynamic setting. These results indicate that TD-TF can be competitive with TD-DMD for favorable initializations, but that the linear delay model remains more robust in this near-linear aerodynamic example. The drag dynamics remain challenging for all models because accurately reproducing sharp extrema requires both amplitude and timing accuracy.

Next, we evaluate the extrapolation capabilities of TD-TF in angle of attack by training on trajectories with $\alpha \in \{-5^\circ, \dots, 8^\circ\}$ and testing on the unseen cases $\alpha = 9^\circ$ and $\alpha = 10^\circ$. For TD-TF, the dependence of the RMSE on the delay length n and hidden dimension h is shown in figure 6. We note that, for TD-TF, we account for initialization variance by averaging the results across 5 training rounds.

For the lift coefficient, TD-TF exhibits large extrapolation errors for small delays ($n \leq 8$), with RMSE values exceeding $1.1 \cdot 10^{-1}$ across all tested hidden dimensions and reaching up to approximately $1.4 \cdot 10^{-1}$. As the delay length increases, the error decreases substantially, with the minimum RMSE of about $3.9 \cdot 10^{-2}$ attained at $n = 13$. Comparable performance is observed for $n = 11$, where the RMSE drops to roughly $4.1 \cdot 10^{-2}$. Increasing the delay length beyond this range does not yield further improvement and can even slightly degrade performance. The RMSE varies noticeably with the hidden dimension h , with the smaller value $h = 50$ tending to yield best performance across delay length.

For the drag coefficient, the extrapolation error decreases rapidly with increasing delay length. For $n \geq 10$, the RMSE falls below $9 \cdot 10^{-3}$ and continues to decrease with n , reaching values close to $5.7 \cdot 10^{-3}$ at $n = 14$ and remaining at the same order for $n = 15$. In this regime, the RMSE shows only weak sensitivity to the hidden dimension h , with variations on the order of 10^{-3} or less across the tested configurations.

For reference, TD-DMD yields RMSE values in the range $[2.5 \cdot 10^{-2}, 9.4 \cdot 10^{-2}]$ for the lift coefficient and $[5.6 \cdot 10^{-3}, 2.3 \cdot 10^{-2}]$ for the drag coefficient over the same set of delay lengths, with the minimum errors attained at the largest tested delay, $n = 15$. This yields, as in the interpolation case, comparative performance with TD-TF. Notably, in the lift prediction extrapolation case, the linear TD-DMD model attained a lower minimum error than TD-TF, which needed a larger n to reduce error. This suggests that when the true behavior with respect to the parameter is close to the linear regime, a linear model can extrapolate more reliably. TD-TF, being more expressive, might need careful regularization or more data to avoid overfitting for extrapolation.



4.3. Case 3: Lorenz'63 system

As the next demonstration case, we examine a canonical chaotic system to assess long-term predictive behavior under strongly nonlinear dynamics.

Setup. The Lorenz'63 system [27] is a classical chaotic model for atmospheric convection, given by

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = x(\rho - z) - y, \\ \frac{dz}{dt} = xy - \beta z, \end{cases} \quad (31)$$

with $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. We generate $N = 1000$ trajectories using $\Delta t = 10^{-2}$ over $[0, 100]$. To remove transients, we discard the first half of each trajectory, which leaves us with trajectories with $K = 5000$ timesteps. We select 900 trajectories for training, and leave the remaining 100 for testing. We retain only the x -component of each trajectory, select $J = 5000$ bursts and subsample with $\tau = 16$.

Results. For the TD-TF and the TD-MLP models, we set their hidden dimension to 100, batch size to 100, and learning rate to 10^{-2} , and we train for 500 epochs. We first assess the performance of the models by considering $n = 1$. Note that, in this case, TD-DMD reduces to the plain DMD, while the TF reduces to a fully-connected residual network [17]. As we can see in figure 7(a), none of the models can accurately reproduce the dynamics.

We increase $n = 3$ in figure 7(b) and notice that TD-DMD does not improve and still converges toward a trivial fixed point near zero. This is to be expected, since it is a fundamentally linear model and cannot represent the nonlinear mechanisms underlying chaotic dynamics. However, TD-TF now captures qualitatively the dynamics of the chaotic attractor. The TD-MLP captures some short-term oscillations, but it does not sustain the alternation between the two lobes of the chaotic attractor and instead settles into an approximately periodic orbit.

To give a more quantitative performance assessment of the TD-TF model on chaotic dynamics, we evaluate whether it reproduces simple statistical signatures of the Lorenz attractor. Specifically, for each test trajectory, we compute (i) the number of *lobe switches*, defined as the number of sign changes in $x(t)$ between time intervals spent in the left and right lobes, and report the corresponding switching frequency obtained by dividing by the trajectory duration, and (ii) *peak statistics* of the x -component, where a “peak” is a local maximum of $x(t)$ and Δt_{peak} denotes the time between successive peaks. We report the mean and standard deviation of these trajectory-wise quantities over $M = 100$ test trajectories. The results are summarized in table 3. TD-TF approximates both the switching rate and the oscillatory peak structure of the true system, although the larger standard deviations indicate substantial trajectory-to-trajectory variability in the long-horizon chaotic rollouts. This variability may be reduced by more systematic choices of initialization, learning-rate scheduling, and regularization. Nevertheless, the qualitative comparison is clear: TD-DMD is nearly non-switching, TD-MLP underestimates the switching rate on average, and TD-TF recovers the lobe-switching dynamics most accurately.

Table 3. Lobe-switching and peak statistics averaged over $M = 100$ test trajectories, for $n = 3$ and $h = 100$. To avoid spurious switches, we count a switch only after the trajectory satisfies $|x(t)| > 0.1$. Means and standard deviations are computed over trajectories.

Model	Lobe switching		Peak statistics	
	Switches	Frequency	Peaks	Mean Δt_{peak}
True	28.56 ± 3.85	0.5721 ± 0.0770	52.05 ± 2.47	0.9565 ± 0.0451
TD-DMD	0.43 ± 0.89	0.0086 ± 0.0177	1.23 ± 1.15	0.7352 ± 0.0700
TD-TF	27.62 ± 13.49	0.5532 ± 0.2701	45.99 ± 8.65	1.1217 ± 0.1721
TD-MLP	8.44 ± 5.71	0.1691 ± 0.1144	65.12 ± 5.03	0.7764 ± 0.0680

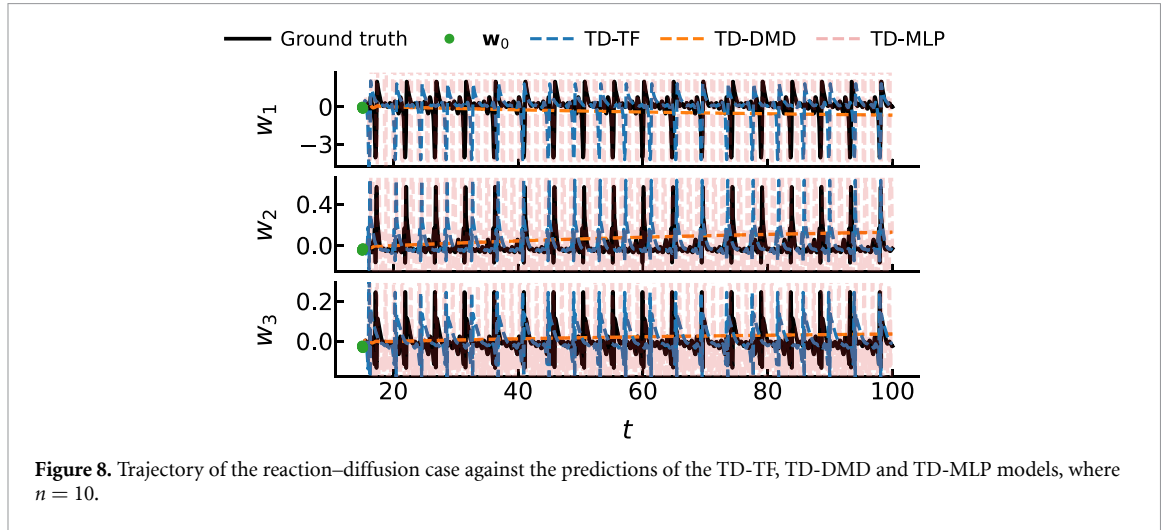


Figure 8. Trajectory of the reaction–diffusion case against the predictions of the TD-TF, TD-DMD and TD-MLP models, where $n = 10$.

4.4. Case 4: Reaction–diffusion partial differential equation

As a final demonstration, we consider a spatially extended nonlinear reaction–diffusion system originally introduced in [13]. This example bridges low-dimensional chaotic dynamics and high-dimensional PDE dynamics and tests the ability of TD-TF to learn effective reduced dynamics obtained from data-driven model reduction.

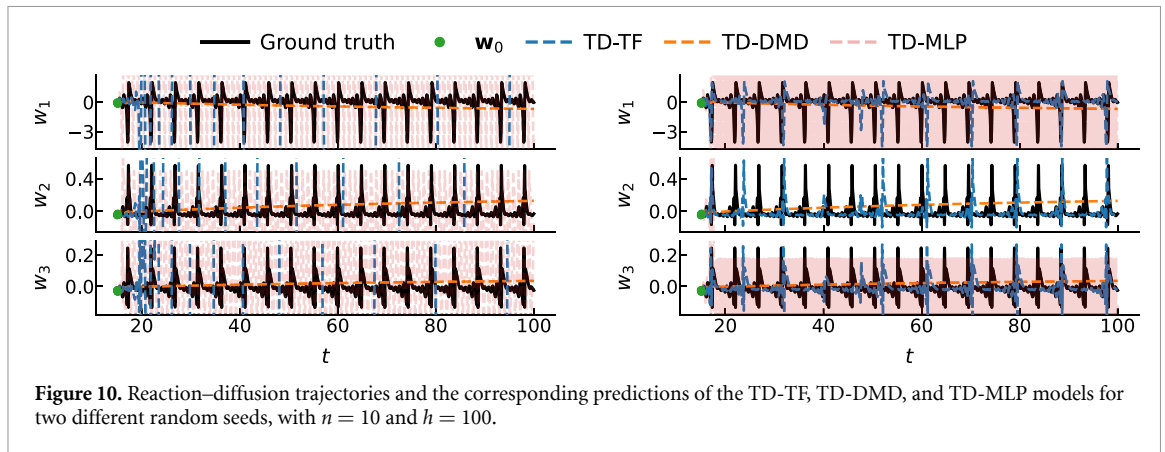
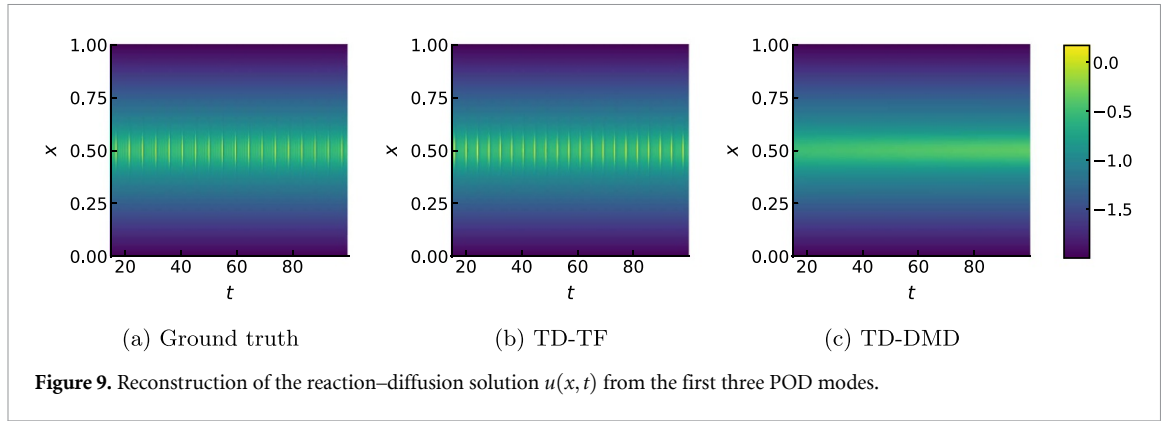
Setup. We consider the following one-dimensional reaction–diffusion system on the spatial domain, $x \in [0, 1]$:

$$\begin{cases} \partial_t u = D \partial_{xx} u + \frac{1}{\varepsilon} (v - u^2 - u^3), \\ \partial_t v = D \partial_{xx} v - u + \alpha, \end{cases} \quad (32)$$

with diffusion coefficient $D = 0.0322307$ and parameters $\varepsilon = \alpha = 0.01$. This system arises as a prototypical activator–inhibitor model, where u acts as a fast activator variable and v as a slower recovery variable. The cubic nonlinearity in u produces bistability and excitable dynamics, while diffusion induces spatial coupling and pattern formation. The chosen parameter values correspond to a regime exhibiting coherent spatio-temporal oscillations and low-dimensional dominant dynamics [13].

We impose Dirichlet boundary conditions $u(0, t) = u(1, t) = -2$, $v(0, t) = v(1, t) = -4$, and solve the PDE using a spectral method [8]. In particular, we discretize in space using a Chebyshev basis with $N_x = 512$ modes and evolve the solution in time using a Runge–Kutta scheme with timestep $\Delta t = 10^{-4}$ up to final time $T = 100$. Data up to time 15 is discarded to remove transient behavior.

From the resulting space–time field $u(x, t)$, we compute a proper orthogonal decomposition (POD) [20]. We retain the first 3 POD modes ($d = 3$), which capture the dominant energetic structures of the dynamics. Projecting onto this basis yields a single trajectory ($N = 1$) of POD coefficients of length



$K = 34000$, which is subsampled in time with $\tau = 5$. We construct $J = 2000$ burst samples using a delay of $n = 10$.

Results. We compare TD-DMD, TD-TF and TD-MLP on the task of long-term prediction of the POD coefficient dynamics. For both neural network models, we use a hidden dimension of $h = 100$, batch size of 100, learning rate of 10^{-2} , and weight decay of 10^{-5} , and training is executed for 500 epochs.

Figure 8 shows the prediction of each POD coefficient for a representative trajectory. TD-DMD exhibits rapid decay and loss of phase information, consistent with the limitations of linear models for nonlinear reduced dynamics, whereas TD-MLP produces high-frequency oscillations with incorrect amplitude and phase. In contrast, TD-TF accurately tracks the oscillatory structure of all three POD modes over long time horizons.

To assess how errors in the reduced dynamics propagate to the physical variables, we reconstruct the solution $u(x, t)$ from the predicted POD coefficients using the retained POD modes. Figure 9 shows the ground truth reconstruction together with the reconstructions achieved by using the TD-TF and TD-DMD models for the same trajectory; TD-MLP is omitted here due to its unstable behavior.

The reconstruction achieved by using the TD-TF model reproduces the dominant spatio-temporal oscillatory patterns observed in the reference solution over the prediction horizon, maintaining comparable amplitudes and spatial organization, though with a clear phase misalignment. In contrast, the reconstruction by TD-DMD exhibits a progressive reduction in oscillation amplitude and a loss of spatial structure over time, consistent with the limitations of linear latent dynamics in representing nonlinear interactions among POD modes.

Results for two additional seeds in figure 10 show that TD-TF is sensitive to random initialization in this setting, with visible variation in rollout quality across seeds, while TD-MLP exhibits similar failure modes across different seeds. Although seed-specific hyperparameter tuning may improve individual runs, improving robustness to initialization remains an important direction for future work.

Summary of the model comparison. The experiments show that TD-TF does not uniformly outperform TD-DMD. TD-DMD remains competitive, and can be preferable, when the dynamics are well approximated by a linear recurrence in delay coordinates, as in the sinusoidal case and parts of the NLR airfoil extrapolation study. TD-TF provides a clearer advantage when the delayed evolution map is nonlinear,

as in the Lorenz system and the reaction–diffusion reduced dynamics. The TD-MLP comparisons indicate that these gains are not solely due to adding a nonlinear map on delay coordinates; the adaptive attention weighting seems to be important in the strongly nonlinear examples.

5. Conclusions

In this work, we have established a theoretical and practical connection between TD linear models such as TD-DMD and the TF architecture. By introducing TD-TF, we have demonstrated that such a simplified TF can be interpreted as a nonlinear extension of TD-DMD, thereby bridging two previously separate modeling paradigms and enhancing interpretability. TD-TF retains the core idea of TD-DMD (each prediction arises from a finite window of past states) while leveraging the expressive power of attention to model nonlinear interactions among those states. This attention mechanism is key to adaptively identifying and weighting the most relevant delayed states for each prediction.

Through numerical experiments on benchmark dynamical systems, we have shown that TD-TF does not simply replace TD-DMD; rather, it complements it. The method is most useful when the delayed evolution map is strongly nonlinear, while TD-DMD remains highly competitive, and sometimes preferable, in linear or weakly nonlinear regimes. These results suggest that TD-TFs might benefit from a design and simplification inspired by their linear counterparts. Our findings highlight that linking classical data-driven models to modern TF architectures can enhance their interpretability. An appealing aspect of the TD-TF approach is its computational efficiency. The architecture uses only $O(n)$ attention operations per step, making training and deployment feasible even as sequence length grows large.

TD-TF represents a step toward trustworthy machine learning for dynamic systems. It combines the transparency of classical delay-embedded models with the power of modern TFs to handle nonlinearity. By respecting the fundamental delay structure, it ensures that only a finite history influences the future and that each past state's influence is explicit. We believe that this approach opens new avenues for data-driven modeling in engineering and science, enabling practitioners to build models that yield powerful yet interpretable predictions.

Future work will explore multi-dimensional and spatially extended systems (e.g. 2D fluid flows using TD-TF on latent variables), as well as incorporating multiple self-attention layers or heads to capture hierarchies of time-scales. Another avenue is enforcing physical constraints in TD-TF, for instance, via symmetry constraints, to further improve physical consistency.

Acknowledgments

The authors thank Giovanni Fantuzzi for valuable conversations. The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR).

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://github.com/albertalcalde/TDTF> [54].

Funding

The work of A. Alcalde was funded by the European Union's Horizon Europe MSCA project ModConFlex (Grant Number 10 107 3558). The HPC system CARA is partially funded by 'Saxon State Ministry for Economic Affairs, Labor and Transport' and 'Federal Ministry for Economic Affairs and Climate Action'.

Author contributions

Albert Alcalde  0009-0000-5147-0406

Conceptualization (equal), Methodology (equal), Software (equal), Validation (equal), Visualization (equal), Writing – original draft (equal)

Markus Widhalm

Conceptualization (equal), Data curation (equal), Supervision (equal), Writing – review & editing (equal)

Emre Yilmaz

Conceptualization (equal), Data curation (equal), Software (equal), Supervision (equal), Writing – review & editing (equal)

References

- [1] Albano E and Rodden W P 1969 A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows *AIAA J.* **7** 279–85
- [2] Arbabi H and Mezic I 2017 Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator *SIAM J. Appl. Dyn. Syst.* **16** 2096–126
- [3] Bao A, Lai J and Gilpin W 2026 Transformers for dynamical systems learn transfer operators in-context (arXiv:2602.18679)
- [4] Bekemeyer P, Ripepi M, Heinrich R and Görtz S 2019 Nonlinear unsteady reduced-order modeling for gust-load predictions *AIAA J.* **57** 1839–50
- [5] Bramburger J J 2024 *Data-Driven Methods for Dynamic Systems* (Society for Industrial and Applied Mathematics)
- [6] Brunton S L, Brunton B W, Proctor J L, Kaiser E and Kutz J N 2017 Chaos as an intermittently forced linear system *Nat. Commun.* **8** 19
- [7] Brunton S L, Proctor J L and Kutz J N 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems *Proc. Natl Acad. Sci.* **113** 3932–7
- [8] Burns K J, Vasil G M, Oishi J S, Lecoanet D and Brown B P 2020 Dedalus: A flexible framework for numerical simulations with spectral methods *Phys. Rev. Res.* **2** 023068
- [9] Calvillo E, Kovachki N B, Levine M E and Stuart A M 2025 Continuum attention for neural operators *J. Mach. Learn. Res.* **26** 1–52 (available at: <http://jmlr.org/papers/v26/24-0879.html>)
- [10] Cao S 2021 Choose a transformer: Fourier or galerkin *Adv. Neural Inf. Process. Syst.* **34** 24924–40 (available at: https://proceedings.neurips.cc/paper_files/paper/2021/file/d0921d442ee91b896ad95059d13df618-Paper.pdf)
- [11] Chen J, Wu K and Xiu D 2025 Due: A deep learning framework and library for modeling unknown equations *SIAM Rev.* **67** 873–902
- [12] Cybenko G V 1989 Approximation by superpositions of a sigmoidal function *Math. Control Signals Syst.* **2** 303–14
- [13] Elezgaray J and Arneodo A 1992 Crisis-induced intermittent bursting in reaction-diffusion chemical systems *Phys. Rev. Lett.* **68** 714–7
- [14] Etkin B 1972 *Dynamics of Atmospheric Flight* (Wiley)
- [15] European Aviation Safety Agency (EASA) 2007 *European Aviation Regulations, Certification Specifications for Large Aeroplanes* (CS-25)
- [16] Geneva N and Zabarar N 2022 Transformers for modeling physical systems *Neural Netw.* **146** 272–89
- [17] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 770–8
- [18] Hemmasian A and Farimani A B 2023 Reduced-order modeling of fluid flows with transformers *Phys. Fluids* **35** 057126
- [19] Herde M, Raonic B, Rohner T, Käppeli R, Molinaro R, de Bézenac E and Mishra S 2024 Poseidon: Efficient foundation models for pdes *Advances in Neural Information Processing Systems* vol 37 pp 72525–624
- [20] Holmes P, Lumley J L, Berkooz G and Rowley C W 2012 *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge Monographs on Mechanics) 2nd edn (Cambridge University Press)
- [21] Hu Z, Daryakenari N A, Shen Q, Kawaguchi K and Karniadakis G E 2025 State-space models are accurate and efficient neural operators for dynamical systems *Neural Netw.* **197** 108496
- [22] Karkaria V, Lee D, Chen Y-P, Yu Y and Chen W 2025 An attention-based spatio-temporal neural operator for evolving physics *Mach. Learn.: Sci. Technol.* **6** 045036
- [23] Koopman B O 1931 Hamiltonian systems and transformation in hilbert space *Proc. Natl Acad. Sci.* **17** 315–8
- [24] Kovachki N, Li Z, Liu B, Azizzadenesheli K, Bhattacharya K, Stuart A and Anandkumar A 2023 Neural operator: Learning maps between function spaces with applications to pdes *J. Mach. Learn. Res.* **24** 1–97 (available at: <http://jmlr.org/papers/v24/21-1524.html>)
- [25] Clainche S L and Vega J M 2017 Higher order dynamic mode decomposition *SIAM J. Appl. Dyn. Syst.* **16** 882–925
- [26] Li Z, Kovachki N B, Azizzadenesheli K, liu B, Bhattacharya K, Stuart A and Anandkumar A 2021 Fourier neural operator for parametric partial differential equations *Int. Conf. on Learning Representations*
- [27] Lorenz E N 1963 Deterministic nonperiodic flow *J. Atmos. Sci.* **20** 130–41
- [28] Loshchilov I and Hutter F 2019 Decoupled weight decay regularization *Int. Conf. on Learning Representations*
- [29] Lu J and Yang S 2025 Linear transformers as VAR models: Aligning autoregressive attention mechanisms with autoregressive forecasting *42nd Int. Conf. on Machine Learning*
- [30] Lu L, Jin P, Pang G, Zhang Z and Karniadakis G E 2021 Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators *Nat. Mach. Intell.* **3** 218–29
- [31] Lütkepohl H 2005 *New Introduction to Multiple Time Series Analysis* (Springer)
- [32] Mezić I 2005 Spectral properties of dynamical systems, model reduction and decompositions *Nonlinear Dyn.* **41** 309–25

- [33] Qin T, Wu K and Xiu D 2019 Data driven governing equations approximation using deep neural networks *J. Comput. Phys.* **395** 620–35
- [34] Schaeffer H 2017 Learning partial differential equations via data discovery and sparse optimization *Proc. R. Soc. A* **473** 20160446
- [35] Schmid P J 2010 Dynamic mode decomposition of numerical and experimental data *J. Fluid Mech.* **656** 5–28
- [36] Schwamborn D, Gerhold T and Heinrich R 2006 The DLR TAU-Code: Recent applications in research and industry *ECCOMAS CFD 2006 Conf., Egmond aan Zee, the Netherlands*
- [37] Solera-Rico A, Vila C S, Gómez-López M, Wang Y, Almashjary A, Dawson S T and Vinuesa R 2024 β -variational autoencoders and transformers for reduced-order modelling of fluid flows *Nat. Commun.* **15** 1361
- [38] Song J, Song Z, Ren P, Erichson N B, Mahoney M W and Li X S 2024 Forecasting high-dimensional spatio-temporal systems from sparse measurements *Mach. Learn.: Sci. Technol.* **5** 045067
- [39] Su J, Ahmed M, Lu Y, Pan S, Bo W and Liu Y 2024 Roformer: Enhanced transformer with rotary position embedding *Neurocomputing* **568** 127063
- [40] Takens F 1981 Detecting Strange Attractors in Turbulence *Dynamical Systems and Turbulence, Warwick 1980* ed D Rand and L-S Young (Springer) pp 366–81
- [41] Tu J H. *Dyn. Mode Decompos.: Theory Appl. PhD Thesis*, Princeton University, 2013
- [42] Box G E P and Jenkins G M 1994 *Time Series Analysis: Forecasting and Control* 3rd edn (Prentice Hall PTR) (<https://doi.org/10.5555/561899>)
- [43] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* p 30
- [44] Verdon J M and Caspar J R 1984 A linearized unsteady aerodynamic analysis for transonic cascades *J. Fluid Mech.* **149** 403–29
- [45] Wang Z, Tao P and Chen L 2026 Delayformer: Spatiotemporal transformation for predicting high-dimensional dynamics *Adv. Sci.* **13** e14671
- [46] Williams M O, Kevrekidis I G and Rowley C W 2015 A data-driven approximation of the koopman operator: Extending dynamic mode decomposition *J. Nonlinear Sci.* **25** 1307–46
- [47] Williams M O, Rowley C W and Kevrekidis I G 2014 A kernel-based approach to data-driven koopman spectral analysis (arXiv:1411.2260)
- [48] Wu H, Xu J, Wang J and Long M 2021 Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting *Advances in Neural Information Processing Systems* vol 34 pp 22419–30
- [49] Wu P, Qiu F, Feng W, Fang F and Pain C 2022 A non-intrusive reduced order model with transformer neural network and its application *Phys. Fluids* **34** 115130
- [50] You H, Fu Y, Wang Z, Yazdanbakhsh A and Lin Y C 2024 When linear attention meets autoregressive decoding: Towards more effective and efficient linearized large language models *41st Int. Conf. on Machine Learning*
- [51] Yun C, Bhojanapalli S, Rawat A S, Reddi S and Kumar S 2020 Are transformers universal approximators of sequence-to-sequence functions? *Int. Conf. on Learning Representations*
- [52] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H and Zhang W 2021 Informer: Beyond efficient transformer for long sequence time-series forecasting *Proc. AAAI Conf on Artificial Intelligence* vol 35 pp 11106–15
- [53] Zwaan R J 1982 Data Set 4: NLR 7301 Supercritical airfoil oscillatory pitching and oscillating flap, NATO AGARD Report 702: Compendium of unsteady aerodynamic measurements
- [54] Alcalde A, Widhalm M and Yilmaz E 2026 Time-delayed Transformer Dataset *TDTF* (available at: <https://github.com/albertalcalde/TDTF>)